UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

DESENVOLVIMENTO DE APLICAÇÃO WEB PARA VISUALIZAÇÃO E ANÁLISE GENÉTICA MICROBIANA

João Miguel Próspero Marques Reis

PROJECTO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2012

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

DESENVOLVIMENTO DE APLICAÇÃO WEB PARA VISUALIZAÇÃO E ANÁLISE GENÉTICA MICROBIANA

João Miguel Próspero Marques Reis

PROJECTO

Projecto orientado pelo Prof. Doutor André Osório e Cruz de Azerêdo Falcão
e co-orientado pelo Prof. Doutor Jorge Manuel Barreto Vítor

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2012

Agradecimentos

Em primeiro lugar manifesto um especial agradecimento ao meu orientador Professor André Falcão e ao co-orientador Professor Jorge Vítor, ao Doutor Richard Roberts e ao New England Biolabs pelo facto de ter sido possível a realização deste projecto. Seguidamente agradeço também toda a ajuda e tempo que dispuseram para que fosse possível uma boa execução do projecto quer a nível pessoal como a nível académico.

Agradeço aos meus pais, irmã e familiares mais próximos por todo o apoio e motivação que me deram ao longo do meu percurso académico com especial destaque nestes dois anos de Mestrado pois sem eles nada disto seria possível.

À Patrícia deixo um agradecimento também especial por toda a paciência, disponibilidade, preocupação e afecto demonstrado principalmente nos momentos menos bons.

Agradeço a todos aqueles que de uma maneira ou de outra fizeram parte da minha vivência académica, com especial destaque para o Alentejano, Monteiro, Fred, Rasteirinho, Reboxo, Açoreano, Faísca, Cigano, Gato, Avô, Insta, Luís, Ramos, Tap, Mariana, Sara, Ana e Xorti. A todos os que frequentam a casa do gordo e meus companheiros de noitadas, o Gordinho, Diogo, Ricardo, Ficalho, Teixeira, Fernandinho, Xico e Mário. Ao LaSIGE e a todas as pessoas que partilharam o mesmo espaço de trabalho comigo, pelas ideias e bons momentos passados um muito obrigado.

Aos meus colegas do Barreiro, minha terra. Entre eles, o Paulinho, Filipe, Sota, Quim, Fábio, Danny, Leite, Diogo, Gonçalo, Bia, Pinto, Saraiva, Jorge Marques, Jorge Cardoso, Ruben, Liliana e aos meus amigos de infância do Beco.

Dedico esta tese à minha sobrinha Matilde.

Resumo

Com o realizar desta tese pretende-se o desenvolvimento de um sistema de informação (GIN - *Genome Inspector*) com base numa arquitectura *web* para análise genómica de bactérias. A aplicação desenvolvida deve ser capaz, com base nos ficheiros com as sequências genómicas anotadas de bactérias e *archaea*, de permitir a criação de mapas genómicos globais individuais, circulares e/ou lineares, onde estão colocados em evidência alguns genes, estando claramente assinalada a sua posição no genoma e orientação. Permite também a criação de mapas genómicos globais múltiplos de modo a ter a comparação visual de vários mapas individuais. Com base nas ferramentas disponiveis no GIN será possível trabalhar o resultado fornecido em programas de alinhamentos múltiplos. Esses alinhamentos também são possíveis de fazer na nossa aplicação. Com esse mesmo resultado também será possível perceber qualquer alteração genómica entre as várias bactérias anotadas nos ficheiros por nós criados.

Todo o desenvolvimento deste sistema de informação foi feito de raiz e envolveu um processamento de formatos de dados complexos, procedimentos de organização de dados para o uso de ferramentas especificas (Ex: BLAST, MUSCLE) com requisitos rigorosos de dados e ao mesmo tempo uma base de dados relacional coerente. O Desenvolvimento *web* desta aplicação envolveu o uso de complexos procedimentos gráficos e modelos de interação que exigiram uma forte ênfase na aplicação de tecnologias HTML5 e AJAX. pretende-se uma continuidade e evolução, uma vez que este sistema demonstrou ser bastante úti para a utilização na resolução de problemas na área de análise genómica de bactérias.

Palavras-chave: Sistemas de informação, Bioinformática, Aplicações *web*, Genómica, Análise de sequências

Abstract

With this work it is intended to describe the development process of a web based information system for baterial genomic analysis (GIN - Genome Inspector). The developed application should be able to organize and retrieve annotated sequence DNA data for several genomes and display the respective genetic maps, circular or linear, where the respective genomic position and orientation is displayed. The application also allows the display of multiple maps for simultaneous comparison. These results can be futher analised through multiple alignment procedures, also made available. With these analysis tools it will be possible to distinguish and determinate genomic changes between different genomes. This system was developed from the ground up and involved the processing of complex data formats, data organization procedures for the usage of specific bioinformatics tools (e.g. BLAST and MUSCLE) with strict data requirements, while simultaneously maintaining a coherent relational database necessary for the remaining application. The web development of the application involved the use of complex graphical procedures and interaction models that required a strong emphasis on the application of AJAX and HTML 5 technologies.

Keywords: Information systems, Bioinformatics, Web aplications, Genomics, Sequence analysis

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

Este relatório pretende introduzir o projecto que será desenvolvido na disciplina Projecto de Engenharia Informática de 2º ano de Mestrado em Engenharia Informática da FCUL (Faculdade de Ciências da Universidade de Lisboa), na área de especialização em SI (Sistemas de Informação).

Este projecto foi desenvolvido ao longo do ano, em parceria com a FFUL (Faculdade de Farmácia da Universidade de Lisboa), objectivando a solidificação de conhecimentos na área de SI (Sistemas de Informação) e aquisição de competências na criação de sistemas de informação disponíveis na *web*. Pretende-se então implementar uma aplicação *web* para análise genética microbiana que permite a criação e visualização de mapas genómicos globais, individuais, circulares ou lineares, que descreverei mais pormenorizadamente nos objectivos do projecto, bem como a realização de alinhamentos.

1.1 Motivação

O primeiro genoma de um ser vivo completamente sequênciado foi publicado em 1995 e demorou um ano a executar. Foi o início de várias revoluções, uma delas ao nível da bioinformática pois a estratégia de sequênciação foi a de fragmentar o genoma em pequenos fragmentos, sequênciar todos e depois utilizar uma aplicação informática para "montar" a sequência completa do genoma [12]. Com o desenvolvimento de sequênciadores automáticos de DNA de grande capacidade e de *software* para resolver rapidamente e anotar as sequências obtidas, é hoje possível sequênciar um genoma bacteriano com seis milhões de pares de bases em menos de 24 horas. Estão já sequênciados mais de catorze mil genomas de bactérias e este número deve aumentar exponencialmente nos próximos anos. É então hoje possível fazer genómica comparativa *in silico* e obter informação para, por exemplo, planear a clonagem e expressão de determinados genes com interesse biotecnológico. O desenvolvimento na área de bioinformática tende a expandir-se, sendo importante a interligação entre as áreas de Biologia e de Informática para tratamento de dados e análise dos mesmos [13], deste modo a parte de investigação associada a este tema torna-se fulcral,

uma vez que é necessária uma pesquisa elaborada do tema e de como tratar os dados mais relevantes. Assim, é possível apresentar resultados viáveis e úteis para quem trabalha com este tipo de sistemas com frequência, facilitando assim o seu trabalho.

A maior parte das ferramentas disponíveis para análise genómica de bactérias hoje em dia estão distribuídas em diversas aplicações *web* e aplicações *desktop*. Com a criação deste projecto que tende a facilitar o processo de análise genómica de bactérias, uma vez que se pretende englobar o máximo de informação possível na mesma aplicação, sem ser necessário recorrer a várias aplicações *web* e processos manuais morosos, dando espaço ao utilizador para escolher o que visualizar e como o quer fazer. Pretende-se também ter um nível de detalhe mais elevado comparativamente a outras aplicações de análise genómica de bactérias de modo a que seja possível ter uma melhor percepção da informação que se encontra em cada organismo bacteriano.

O facto de se optar por uma aplicação baseada numa arquitectura *web* facilita o acesso dos utilizadores, uma vez que tudo o que necessitam é de um *browser* compatível com a aplicação e as suas componentes. Uma vez que a maior parte da carga é processada do lado do servidor, os utilizadores não têm de se preocupar com o espaço em disco para executar a nossa aplicação. Todo o projecto foi realizado de raiz e deste modo tivemos de nos preocupar tanto com a estruturação como a implementação do mesmo. Com isto é possível mostrar e colocar em prática todos os conhecimentos adquiridos ao longo da formação académica e em especial destaque a formação na área de SI.

1.2 Enquadramento

1.2.1 Enquadramento da equipa

Este projecto foi desenvolvido em parceria com a FFUL, tendo a mesma proposto à FCUL a realização de uma aplicação para visualização e análise genómica microbiana enquadrada num projecto de tese de mestrado de Informática na área de SI. Todo o projecto foi realizado no laboratório de investigação do LaSIGE (*Large-Scale Informatics Systems Laboratory*) da FCUL enquadrado na equipa de investigação do XLDB e está alojado nos servidores do XLDB do DI (Departamento de Informática). O projecto em questão foi financiado pelo *New England Biolabs, Inc. (Ipswich, MA - USA)*.

1.2.2 Enquadramento do problema

Um dos problemas que se coloca hoje em dia a nível de aplicações para análise microbiana e visualização de mapas genómicos microbianos é a diversidade de aplicações existentes e com um nível de detalhe tão baixo que por vezes se torna complicado analisar a informação pretendida, normalmente um gene, no meio de toda a informação apresentada, deste modo a nossa aplicação tende a colmatar essa falha. Pretende-se que a nossa

aplicação centralize só a informação necessária ao utilizador alvo numa única aplicação e com um nível de detalhe adequado e de fácil compreensão. Deste modo oferecemos várias opções de vista e várias alternativas de pesquisa para o mesmo problema, como descritas na secção 1.3.

1.3 Objectivos

Pretende-se criar um sistema de informação *web*, para visualização de genomas microbianos, de modo a efectuar a pesquisa de sequências genómicas através de um determinado gene de um organismo, através de uma região indicada pelo utilizador (inicio e fim da sequência de um determinado organismo) ou através de uma sequência inserida directamente pelo utilizador na aplicação ou de um ficheiro externo. Estas pesquisas são feitas tendo como base genomas completamente anotados em repositórios públicos (GenBank).

É necessário ter vários modos de visualização para a mesma pesquisa. Temos assim os mapas genómicos individuais circulares e/ou lineares, onde estão colocados em evidência alguns genes/regiões estando claramente assinalada a sua posição no genoma e orientação, os alinhamentos simples e múltiplos de sequências, onde é possível verificar as várias alterações às mesmas. Será também possível a visualização dos genes fundamentais, RNA 5S, RNA 16S, RNA 23S ou tRNAs nos mapas circulares e a criação de mapas genómicos múltiplos permitindo a comparação visual de vários mapas individuais.

Então, mostrou-se necessário criar uma aplicação web e toda a sua arquitectura para visualização e análise genética microbiana. A nível estrutural foram criados um *backend* e um *frontend user friendly*, de modo a que fosse de fácil percepção e fácil interpretação. Estes estão descritos mais pormenorizadamente no capítulo 5.

1.4 Metodologia

A estratégia de desenvolvimento para este projecto foi centrada num modelo iterativo e incremental. Esta escolha deve-se ao facto de este ser o modelo mais adequado para projectos longos e que possam estar sujeitos a alterações de requisitos no decorrer do projecto, deve-se também ao facto de existir a possibilidade de avaliar os riscos e pontos críticos do projecto e identificar medidas para os eliminar ou controlar.

Neste modelo define-se o projecto em pequenas partes das quais vão resultar incrementos de trabalho. Dividimos o nosso projecto/aplicação em várias fases distintas, destacando-se a Familiarização, Estrutura da Aplicação, *Backend*, *Frontend*. O *Frontend* foi dividido em quatro sub-itegrações, alinhamento simples, gráficos circulares, gráficos lineares e alinhamentos múltiplos. Cada uma das iterações passam por várias fases e se no final das mesmas se se verificar uma conclusão positiva então pode-se passar à iteração

seguinte caso contrário volta-se à primeira fase dessa mesma iteração e assim consecutivamente tal como mostra a figura 1.1.

1.5 Desafios tecnológicos

Como desafios tecnológicos deparamo-nos com o facto de ter de se criar uma aplicação *web* em que fosse compatível com o maior número possível de *browsers* e com o uso indeterminado de utilizadores alvo. Tivemos de integrar várias linguagens de programação em simultâneo e ter um serviço que fosse capaz de responder aos pedidos dos utilizadores. Neste momento já nos apercebemos que com a criação de bases de dados muito grandes o tempo de resposta não é de facto muito rápido, no entanto é algo que será possível ser restruturado e optimizado futuramente.

O facto de usarmos determinados programas, que utilizam ficheiros com um formato próprio, de leitura de informação, necessitamos de ter informação replicada tanto nesses ficheiros como na nossa base de dados e a elevada dimensão de informação implica ter ficheiros também demasiado grandes (*gigabytes*). Com isto a sua leitura também se torna lenta e isso transparece para o utilizador.

1.6 Plano de trabalho

As fases de planeamento do plano de trabalho foram todas cumpridas tal como descritas no mapa de *gantt*, figura 1.2. Existiu um atraso a nível de tempos devido às fases de optimizações e revisões do projecto, assim foi necessário mais tempo para essas fases que fez com que a escrita do relatório começasse mais tarde do que o previsto.

1.7 Organização do documento

Este relatório está dividido em seis capítulos:

- Capítulo 2 – São apresentados alguns conceitos chave para a realização do nosso projecto, como conceitos de Bioinformática e conceitos de desenvolvimento de aplicações *web*.

- Capítulo 3 – Neste capítulo é apresentada informação referente a trabalhos já realizados na área do nosso projecto e com funções semelhantes às que foram realizadas.

- Capítulo 4 – Descreve um conjunto de métodos para a análise do problema em questão, tal como a análise de requisitos, casos de uso, diagrama de classes, e modelo de actividades.

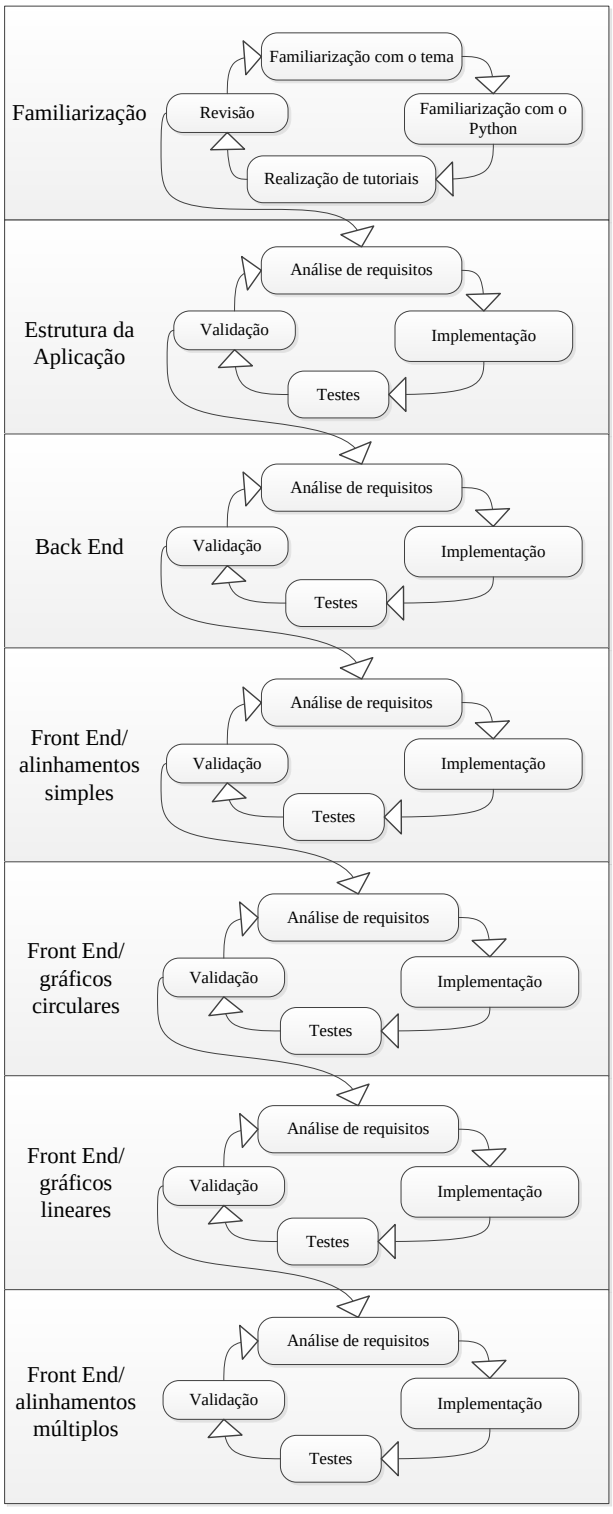


Figura 1.1: Ciclo de vida do projecto.

ID	Task Name	Start	Finish	Duration
1	Familiarização com o problema	03-10-2011	28-10-2011	20d
2	Familiarização com a técnologia	17-10-2011	28-10-2011	10d
3	Estruturar aplicação	31-10-2011	11-11-2011	10d
4	Implementação da aplicação	14-11-2011	29-06-2012	165d
5	BackEnd	14-11-2011	16-12-2011	25d
6	FrontEnd	19-12-2011	15-06-2012	130d
7	Alinhamento Simples	19-12-2011	13-01-2012	20d
8	Gráficos circulares	17-01-2012	02-03-2012	34d
9	Gráficos lineares	05-03-2012	06-04-2012	25d
10	Alinhamentos múltiplos	09-04-2012	27-04-2012	15d
11	Revisão/optimização	30-04-2012	25-05-2012	20d
12	Testes de usabilidade	28-05-2012	18-06-2012	16d
13	Escrita do relatório	28-05-2012	29-06-2012	25d

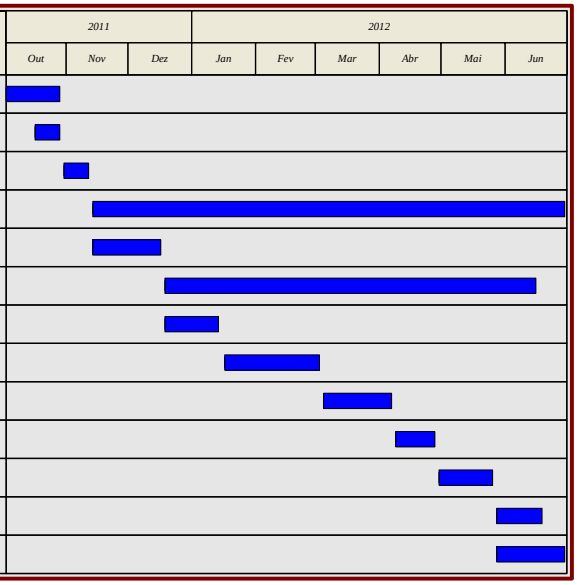


Figura 1.2: Mapa de gantt.

- Capítulo 5 - Trabalho realizado. Descreve todas as etapas realizadas ao longo deste projecto, o modo como está dividido, quais as linguagens usadas e quais os programas externos que tivemos de usar e como foram usados. Descrevemos também o modelo escolhido para a realização dos testes de usabilidade.
- Capítulo 6 – Capítulo de conclusão relativamente ao trabalho realizado, às dificuldades encontradas ao longo do projecto e qual o trabalho futuro a realizar.

Capítulo 2

Análise do problema

2.1 Definição e análise de requisitos

A análise de requisitos é a primeira fase do processo de desenvolvimento de *software*. É nessa fase que nos sentamos e conversamos com o nosso cliente e, é também nessa fase que conheceremos o problema que precisamos resolver. É importante conhecer bem o problema para que consigamos desenvolver um *software* de qualidade e, para isso, precisamos questionar o cliente. Nesse momento não deve existir a preocupação em como algo será feito, precisamos apenas de saber o que é preciso fazer para satisfazê-lo.

Ao prepararmo-nos para conversar com o cliente, devemos ter o cuidado de deixá-lo à vontade. É fundamental que o cliente esteja disposto a conversar naquele momento, caso contrário, não seremos capazes de compreender o problema e, consequentemente, não teremos um *software* que o satisfaça e não teremos um *software* de qualidade.

2.2 Enquadramento

É elaborada uma lista de questões (*checklist*) dirigidas ao problema, de modo a tentar perceber todas as respostas ao problema em questão e de poupar tempo na resolução do mesmo. Foram colocadas algumas questões chave de modo a determinar o que será realmente necessário fazer e de que modo. Iremos descrever as questões colocadas e como influenciaram a realização da nossa aplicação *web*.

- Quem é o cliente?
- Qual o problema?
- Quem utilizará o *software*/aplicação?
- Utilizadores diferentes têm necessidades diferentes?
- Existem aplicações semelhantes que podem ser usadas como referência?

Quem é o cliente?

O nosso cliente é a FFUL e tem como destinatários todos os investigadores da área de microbiologia, é suposto a aplicação ser utilizada por várias Faculdades de vários países na área da genética microbiana. O financiamento foi feito pelo *New England Biolabs, Inc. (Ipswich, MA – USA)*.

Qual o problema?

Os utilizadores têm de ter a possibilidade de aceder à visualização de mapas genómicos (circulares e/ou lineares) individuais e múltiplos, à visualização de alinhamentos simples e à criação de alinhamentos múltiplos. Também terá de ser possível ver os tRNA, rRNA 5s, rRNA 16s e rRNA 23s nos mapas genómicos circulares para servirem, se necessário, de marcadores internos. A pesquisa de sequências pode ser feita tanto pelo gene de um determinado organismo como por uma sequência pré definida pelo utilizador (início e fim) de um determinado organismo, ou por uma sequência inserida manualmente pelo utilizador, quer seja directamente na aplicação quer através de um ficheiro externo com a sequência pretendida. Deverá ser mostrada toda a informação relevante referente à pesquisa feita pelo utilizador, como o nome dos organismos pesquisados, o início e o fim das sequências encontradas através do alinhamento simples, entre outras. Também tem de ser dada a possibilidade de fazer *download* dos mapas genómicos circulares individualmente.

O administrador tem de ter a liberdade para criar as bases de dados, inserir os organismos e toda a informação correspondente aos mesmos, apagar um organismo e apagar uma base de dados através do nome da mesma.

Quem utilizará o *software*/aplicação?

A nossa aplicação será essencialmente usada por investigadores na áreas de biologia molecular e bioinformática. Existirá um *user guide* e várias ajudas ao longo da aplicação de modo a facilitar a sua utilização e um espaço específico para enviar comentários para o administrador da aplicação.

Utilizadores diferentes têm necessidades diferentes?

Existem vários tipos de utilizadores e deste modo a nossa aplicação visa preencher todos os requisitos dos mesmos, iremos fornecer a opção de visualizar separadamente os vários tipos de mapas ou alinhamentos bem como os diferentes tipos de RNA. As bases de dados poderão ser criadas individualmente de modo a que cada utilizador tenha uma base de dados própria por um tempo determinado pelo administrador da aplicação.

Existem aplicações semelhantes que podem ser usadas como referência?

Existem várias aplicações que podem ser usadas como referência, e outras que foram usadas por nós, como as explicadas no capítulo 3. No entanto e tendo em conta os utilizadores alvo, foram definidas especificações próprias de modo a satisfazer as questões propostas e a facilitar a visualização dos resultados obtidos.

2.2.1 Requisitos funcionais

Requisitos funcionais são aqueles que definem como o sistema deve ou não reagir. É onde definimos o que uma entrada específica do utilizador causará como consequência no sistema.

Frontend

- Pesquisa por gene.
  - *Input* da base de dados
  - *Input* do organismo em questão
  - *Input* dos organismos para o alinhamento
  - *Input* do gene
  - *Input* do tipo de RNA
- Pesquisa por região.
  - *Input* da base de dados
  - *Input* do organismo em questão
  - *Input* dos organismos para o alinhamento
  - *Input* da região inicial da sequência (*start*)
  - *Input* da região final da sequência (*end*)
  - *Input* do tipo de RNA
- Pesquisa por sequência.
  - *Input* da base de dados
  - *Input* da sequência (manual ou ficheiro externo)
  - *Input* dos organismos para o alinhamento
  - *Input* do tipo de RNA
- Visualização dos alinhamentos simples.

- Visualização dos gráficos circulares.
- Visualização dos gráficos lineares.
- Realização de alinhamentos múltiplos.
  - *Input* do início da sequência (*start*)
  - *Input* do final da sequência (*end*)
  - *Input* do formato de saida

Backend

- Criar nova base de dados.
  - *Input* do nome da base de dados
- Apagar base de dados.
  - *Input* do nome da base de dados
- Actualizar base de dados.
  - *Input* do nome da base de dados
- Introduzir genomas na base de dados.
  - *Input* do nome da base de dados
  - *Input* do código GenBank do organismo
  - *Input* do ficheiro GenBank do organismo
- Apagar genomas na base de dados.
  - *Input* do nome da base de dados
  - *Input* do nome do organismo

2.2.2 Requisitos não funcionais

Os requisitos não funcionais são aqueles que descrevem as qualidades do sistema como usabilidade, privacidade, segurança, desempenho, etc...

- Privacidade - Nenhum utilizador poderá inserir ou remover organismos da base de dados e criar ou remover bases de dados. Só o administrador tem essas permissões através de um *login*.

- Segurança - O *login* por parte do administrador será feito através de um *username* e *password*, sendo esta última cifrada antes da verificação na base de dados MySQL.
- Desempenho - Toda a aplicação será desenvolvida de modo a que seja possível responder a vários pedidos por parte dos utilizadores e de modo a distribuir o peso do pedido tanto pelo lado do servidor (bases de dados) como pelo lado do cliente (criação dos gráficos).
- Compatibilidade - A aplicação deverá ser compatível em qualquer plataforma (Windows, Linux, Mac) utilizando qualquer *browser* que suporte o elemento *canvas* e HTML5.

2.2.3 Casos de uso

Um caso de uso descreve um objectivo que o utilizador pretende alcançar. Esse objectivo é sempre concreto e especifica as exigências para se alcançar esse objectivo. Toda a actividade que ocorre para de facto se alcançar esse objectivo não é importante para o cliente (utilizador). Na figura 2.1, mostramos o nosso modelo de casos de uso.

2.2.4 Modelo de actividades

Um modelo de actividades é como um fluxograma, onde uma actividade implica a outras, condições levam a outras actividades, falhas geram excepções até que, no final, um determinado objectivo tenha sido alcançado, uma excepção tenha sido disparada ou um aviso enviado ao actor (nesse caso, o actor pode ser o utilizador, uma classe ou uma operação qualquer).

Passamos a mostrar os modelos de actividades, um para a pesquisa por gene e outro para a pesquisa por região e por sequência a nível de *Frontend*. A nível de *Backend* temos um modelo de actividades para a inserção de organismos através de códigos GenBank ou através de um ficheiro externo com o organismo pretendido em formato GenBank.

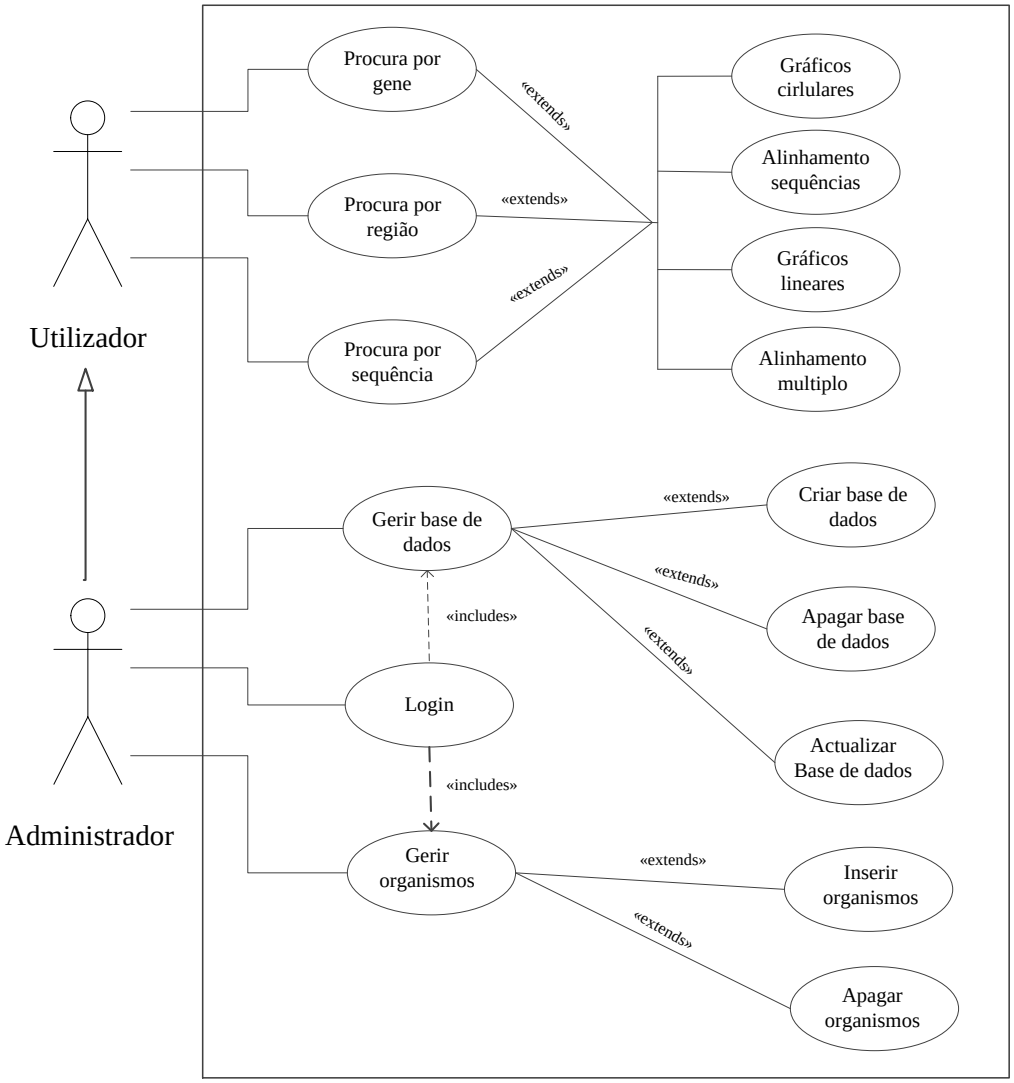


Figura 2.1: Casos de uso.

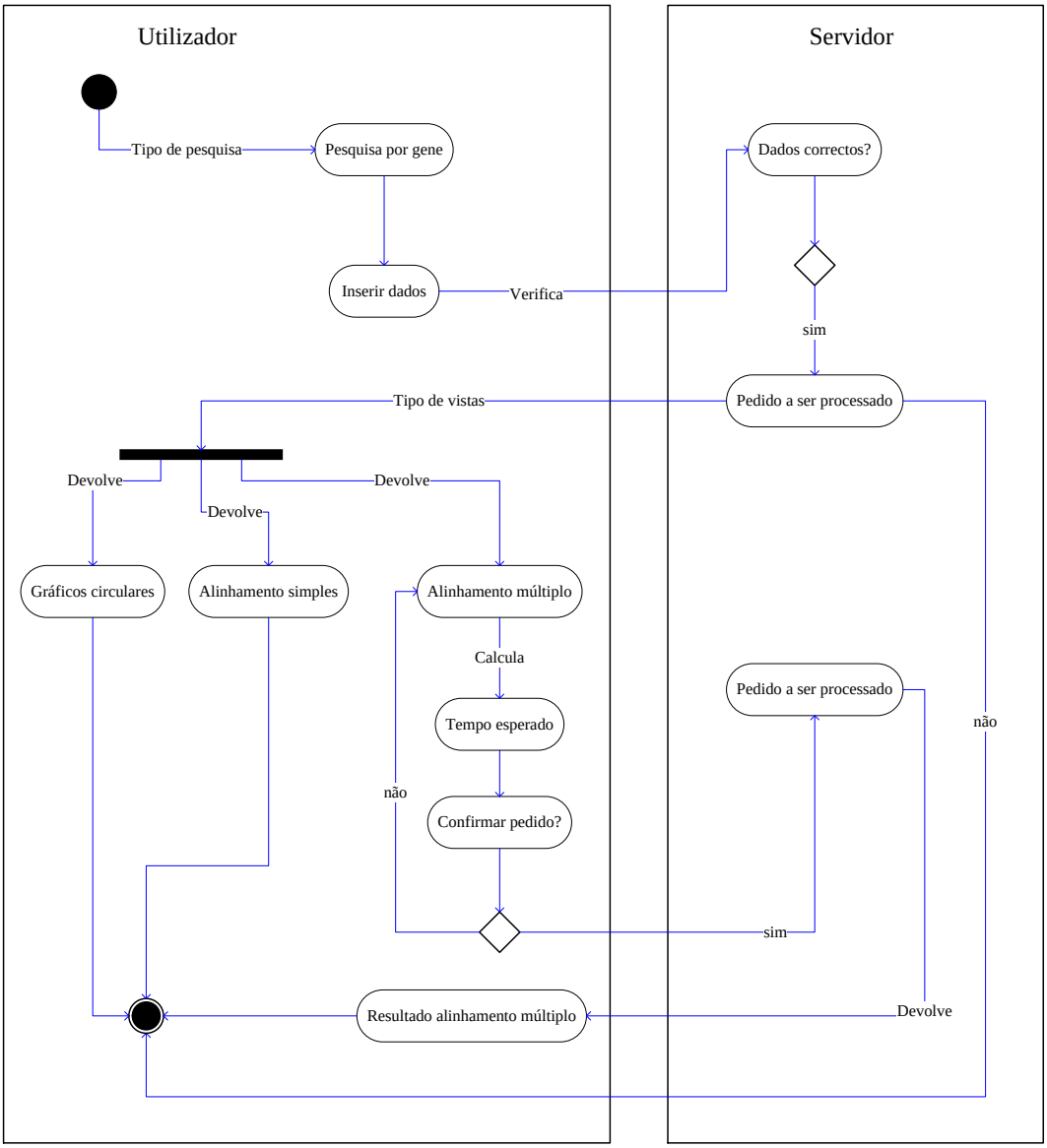


Figura 2.2: Modelo de actividades da pesquisa por gene.

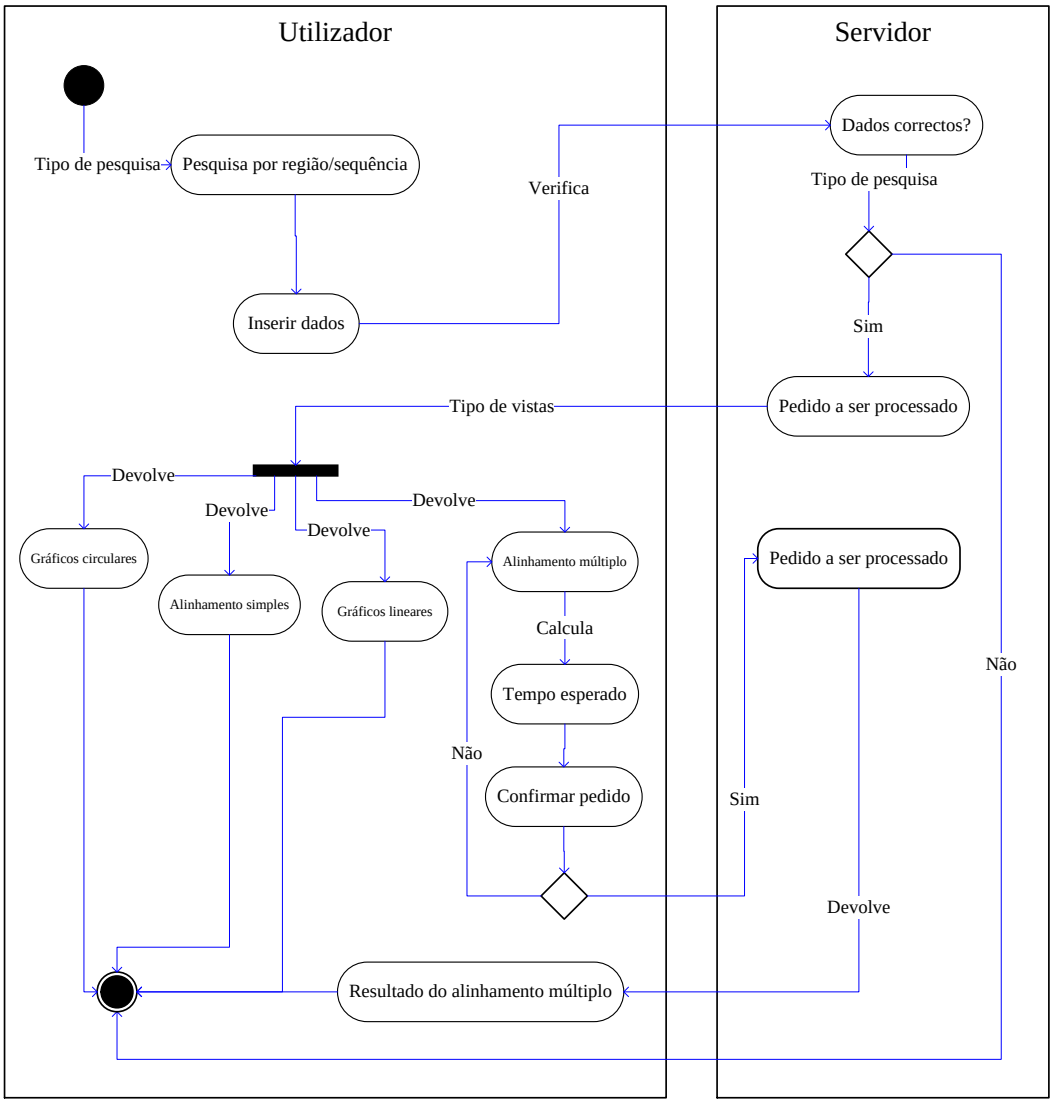


Figura 2.3: Modelo de actividades da pesquisa por região e sequência.

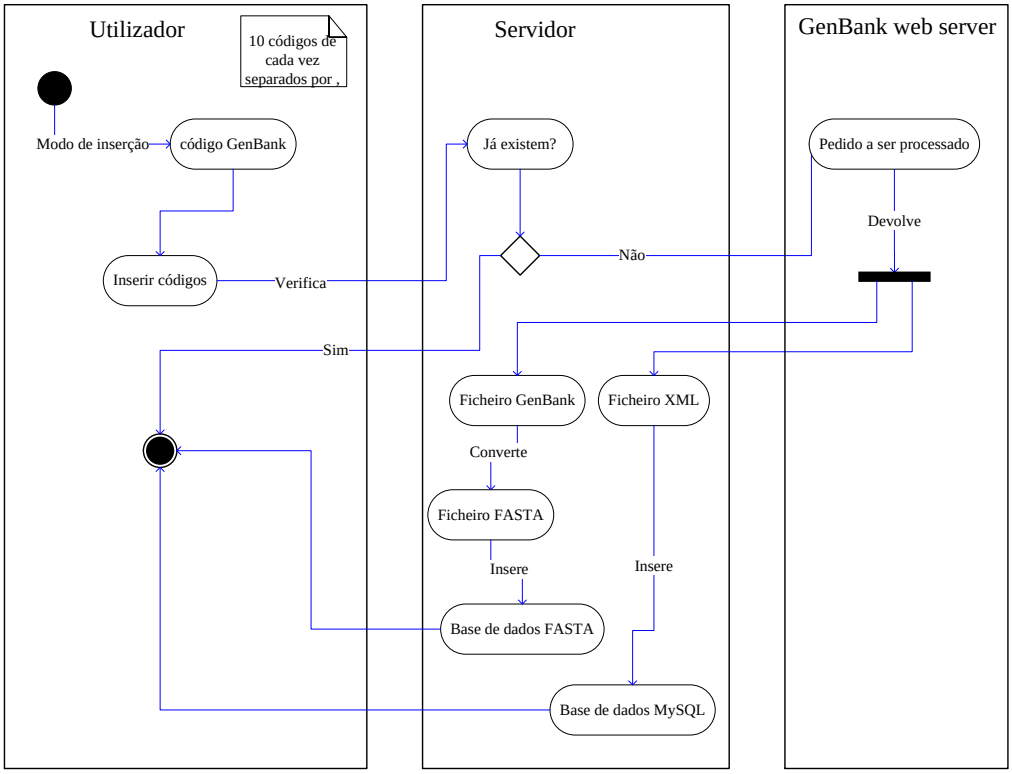


Figura 2.4: Modelo de actividades da inserção de organismos por código GenBank.

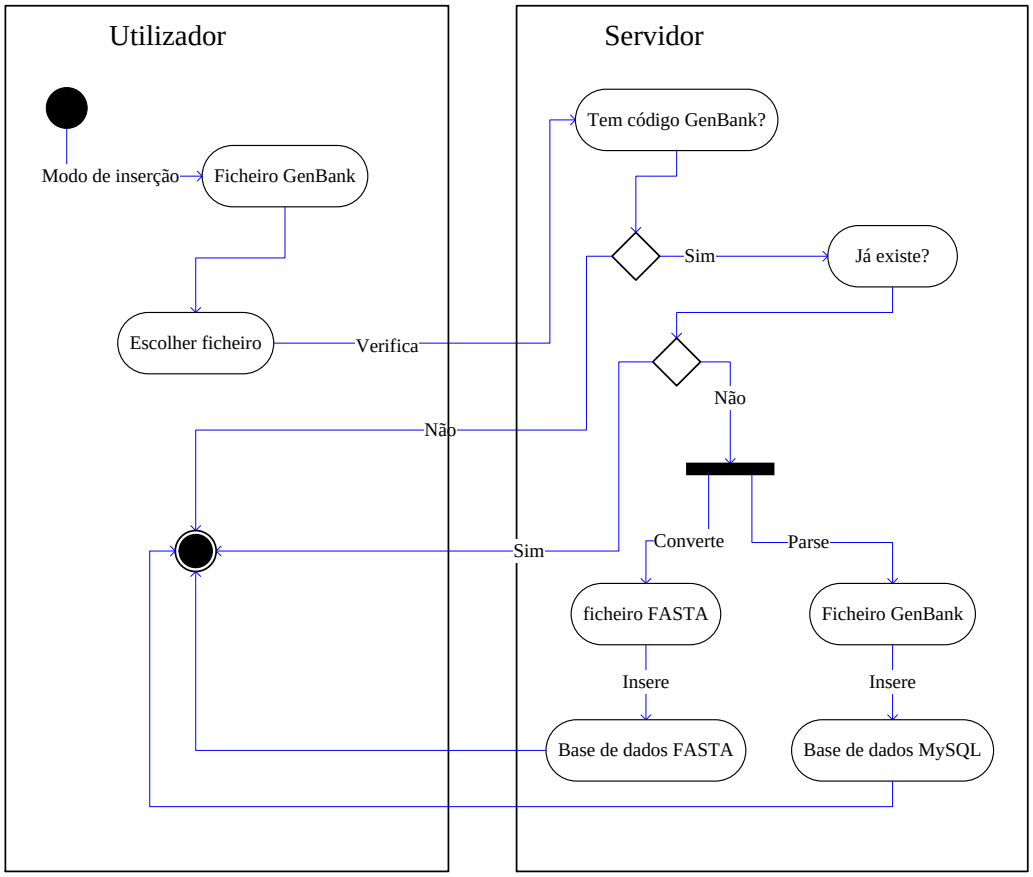


Figura 2.5: Modelo de actividades da inserção de organismos por ficheiro GenBank.

Capítulo 3

Trabalho relacionado

3.1 REBASE

A REBASE [24] é uma base de dados abrangente onde se encontra informação sobre as enzimas de restrição, DNA metiltransferase e proteínas relacionadas, envolvidas no processo biológico de restrição-modificação (RM). Contém informações devidamente referenciadas sobre zonas de reconhecimento e clivagem. Contém também descrições completas do conteúdo dos sistemas RM onde todos os genomas estão sequênciados [16]. A REBASE inclui a visualização gráfica dos sistemas de restrição nos genomas completamente sequênciados e um acesso à informação dos mesmos referenciados no GenBank. Os sistemas RM são classificados actualmente em quatro tipos, tipo I, II, III e IV. Tanto os gráficos circulares como os lineares mostram todos os tipos de sistemas RM existentes em cada organismo bem como as proteínas neles contidos. As proteínas de cada tipo nem sempre são contínuas e nesse caso também são colocadas as proteínas não identificadas a uma cor específica (cinzento) podendo estas estar alternadas com as proteínas correspondentes a cada tipo. As proteínas listadas são proteínas fundidas, ou seja, são subtipos que contêm duas proteínas juntas.

A visualização da informação do organismo pode ser visualizada através de uma referência para o GenBank e neste é possível visualizar toda a informação referente ao organismo, bem como toda a sua sequência genética. Os gráficos circulares representam o genoma do microrganismo, mesmo quando este é linear, com as posições dos tipos de sistemas RM encontrados, cada tipo é representado por uma cor diferente e tem contido em forma de setas todas as proteínas encontradas e as direcções desse mesmo tipo, cada seta (proteína) também tem uma cor específica associada, tal como mostra a figura 3.1.

Os gráficos lineares mostram numa forma mais específica e por tipo todos os sistemas RM encontrados bem como informação mais detalhada das suas posições e características, tais como uma *flag* a identificar os motivos proteicos conservados das metiltransferase, cada proteína também é representada em forma de seta e com uma cor específica, tal como mostra a figura 3.2. Ainda é possível ao clicar no nome de cada proteína obter uma

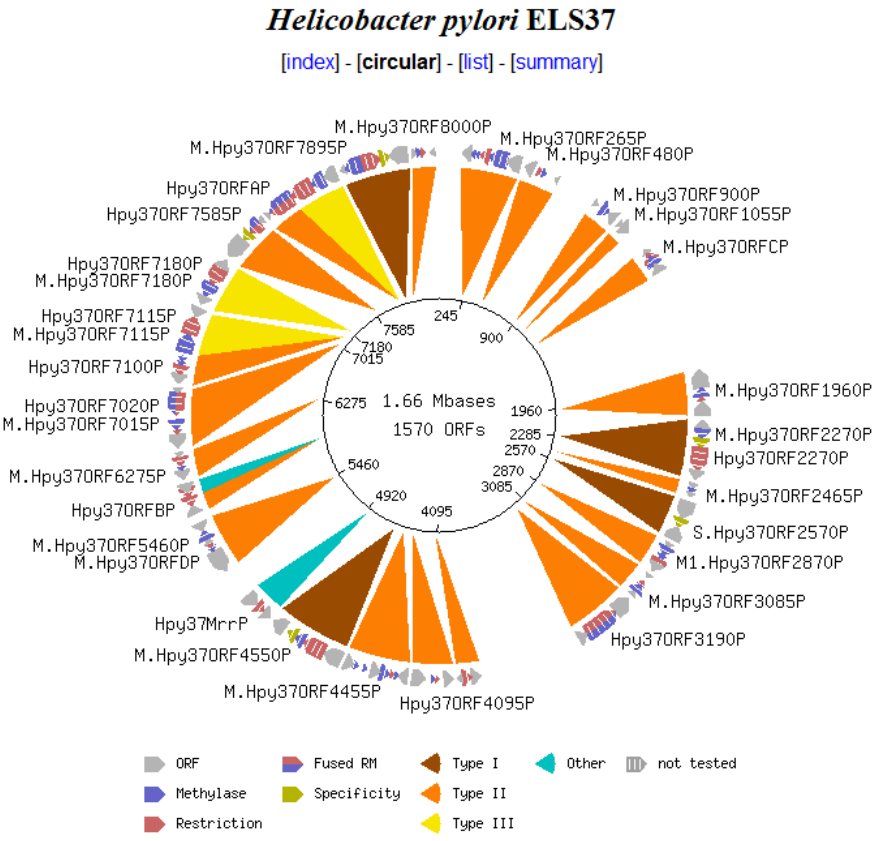


Figura 3.1: Gráfico Circular da *Helicobacter pylori ELS37*. *(http://tools.neb.com/~vincze/genomes/view.php?view_id=20372)*

informação mais detalhada da mesma, como a descrição, o *locus tag* e o código GenBank do organismo, figura 3.3

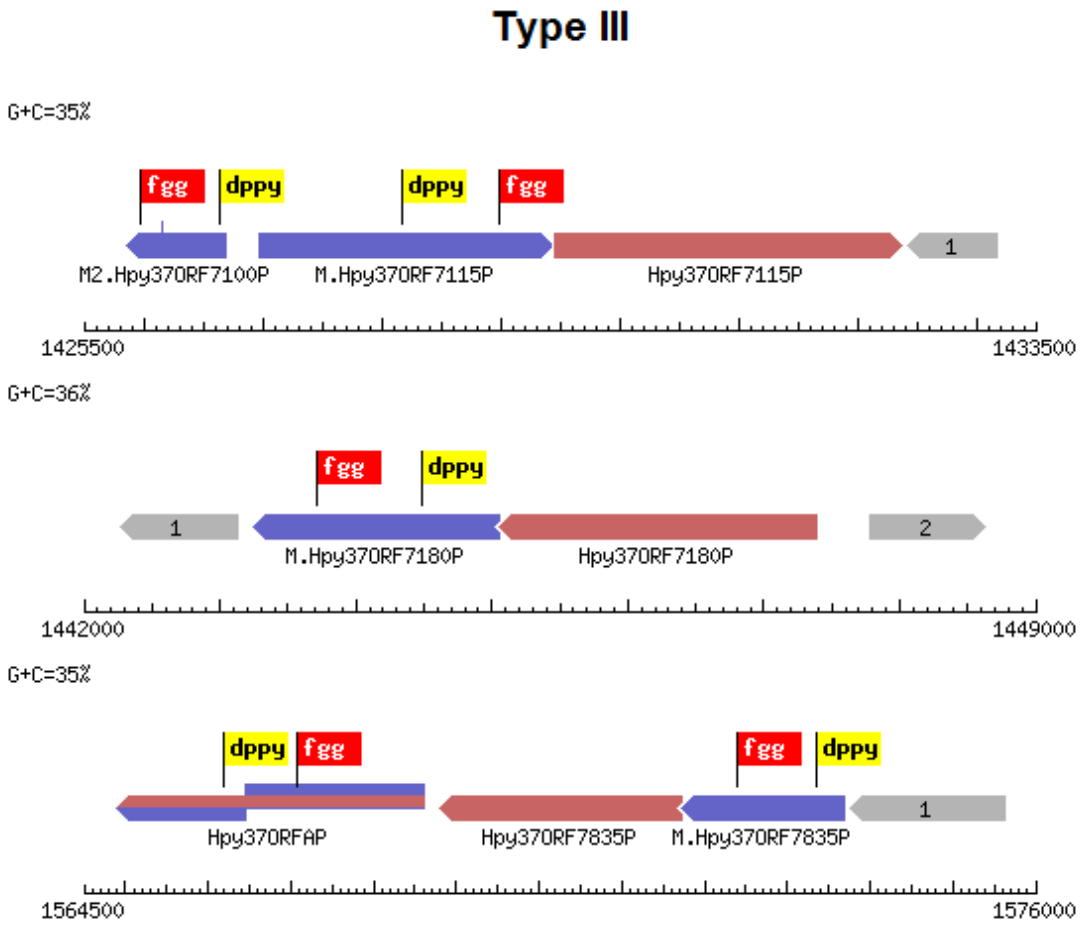


Figura 3.2: Gráfico Linear tipo III da *Helicobacter pylori ELS37*.
(http://tools.neb.com/~vincze/genomes/view.php?seq_id=21198&list=1)

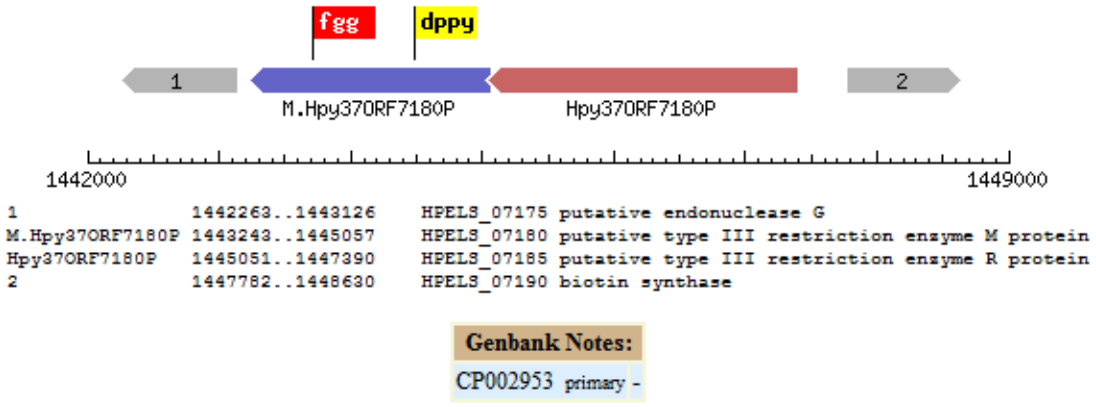


Figura 3.3: Gráfico Linear detalhado do M.Hpy370RF7180P da *Helicobacter pylori ELS37*.
(http://rebase.neb.com/cgi-bin/seqget?M.Hpy37ORF7180P)

Capítulo 4

Conceitos e Tecnologias envolvidas

4.1 BioInformática

A bioinformática é um ramo da ciência biológica que lida com o estudo de métodos para armazenamento, recuperação e análise de dados biológicos, tais como ácidos nucleicos (DNA/RNA) e sequências de proteínas, estrutura, função, caminhos e interações genéticas. Também lida com algoritmos, bases de dados e sistemas de informação, tecnologias *web*, inteligência artificial e da teoria da informação, biologia estrutural, engenharia de *software*, entre outras. Bioinformática é uma ciência que veio colmatar a necessidade de compreender certas funções biológicas à aplicação das técnicas de Informática, no âmbito da análise da informação na área de estudo da Biologia de sequências biológicas [18] [15].

Existem várias ferramentas Bioinformáticas que permitem a resolução de problemas complexos como perceber se determinada sequência de um organismo é semelhante a outro organismo e onde é que se encontra, ou ser possível realizar alinhamentos múltiplos num período de tempo aceitável.

4.1.1 Organização de sequências biológicas

Existem várias ferramentas de alinhamentos de sequências genómicas, no entanto todas essas ferramentas necessitam de ler e entender essas sequências. Para resolver essa questão existem formatos próprios de armazenamento de sequências, de modo a que as mesmas possam perceber quais as sequências existentes e distingui-las entre si.

Ficheiros FASTA

FASTA [20] é um formato para representar sequências de nucleótidos, também permite representar sequências de nomes e comentários precedendo às mesmas. O formato FASTA tem a característica de ser um formato de ficheiro de texto começado pelo símbolo > e seguido de uma descrição ou comentário e nas linhas seguintes a sequência pretendida [20]. É o formato usado para reconhecimento das sequências através das ferramentas

acima descritas. Uma vez que é um formato que pode ser lido através de um *software* para alinhamentos simples, múltiplos, entre outros e foi utilizado para guardar toda a informação das sequências de DNA completas relativas aos organismos a estudar. Criámos um ficheiro FASTA para cada entrada na entidade espécie (base de dados de cada utilizador) onde estão anexados todos os genomas completos de cada organismo existente nessa mesma espécie.

Além de usarmos este tipo de formato para guardar todas as sequências genéticas completas de cada organismo em ficheiros separados com nome de cada espécie, será também usado para criar vários ficheiros temporários com determinadas sequências de aminoácidos. Estes ficheiros vão permitir que o *software* do BLAST e do MUSCLE sejam executados sobre certas sequências guardadas nesse formato como será explicado mais abaixo no capítulo 5.

4.1.2 Genomas de bactérias e anotações

Existem vários genomas de bactérias completamente sequênciados, no entanto esses genomas só estão completos quando são conhecidos todos os seus genes existentes. Neste caso é necessário que esses genes estejam guardados numa base de dados de modo a que se possa verificar mais tarde se um determinado genoma tem todos os genes devidamente guardados e deste modo puder ser anotado.

GenBank

O GenBank [21] é uma base de dados que contém publicamente disponíveis sequências de nucleótidos para mais de 300.000 organismos nomeados ao nível de género ou inferior, obtidos principalmente através de contribuições de laboratórios individuais e submissões em lotes de projectos de sequênciamento em larga escala [2]. Esta base de dados é produzida e mantida pelo *Nacional Center for Biotechnology Information* (NCBI), como parte da *Internacional Nucleotide Sequence Database Collaboration* (INSDC). NCBI é uma parte do *National Institutes of Health* nos Estados Unidos. O GenBank continua a crescer a uma taxa exponencial, estando previsto dobrar a cada 18 meses em 2008 e em cada 30 meses em 2009. As sequências da base de dados GenBank são classificadas e podem ser consultadas através de uma taxonomia baseada em sequências completas. Cada entrada inclui uma descrição concisa da sequência, o nome científico e a taxonomia do organismo de origem, referências bibliográficas e uma tabela de características onde estão listadas áreas de importância biológica, como as regiões de codificação e as suas traduções proteicas, unidades de transcrição, regiões de repetição e os locais de mutações ou modificações [2] [3].

Em mais de 20 anos desde a sua criação, o GenBank tornou-se uma das bases de dados mais importante, cujos dados foram acedidos e citados por milhões de investigadores de

todo o mundo. A nível de estruturação da nossa aplicação é através desta base de dados que iremos obter toda a informação relativa aos organismos pretendidos de modo a serem guardados numa base de dados nossa para tratamento de informação pretendida.

4.1.3 Prospecção de informação e alinhamento de sequências

Um alinhamento de sequências é uma forma de organizar sequências primárias de DNA, RNA ou proteínas para identificar regiões similares que possam ser consequência de relações funcionais, estruturais ou evolucionárias entre elas. Para efectuar alinhamentos de sequências é necessário verificar se uma determinada sequência existe, totalmente ou parcialmente, num ou em vários genomas completamente sequênciados e anotados. De modo a que se perceba quais as diferenças que existem nas sequências encontradas entre os vários genomas é importante que se diga onde e quais as diferenças entre as mesmas e assim consegue-se perceber até que ponto é que duas ou mais sequências são diferentes uma da outra.

BLAST

O BLAST [19] (*Basic Local Alignment Search Tool*) é um algoritmo para comparar informações de sequências biológicas primárias, tais como sequências de aminoácidos de diferentes proteínas ou nucleótidos de sequências de DNA. Uma pesquisa BLAST permite que um investigador compare uma sequência fornecida através de uma consulta a uma biblioteca ou a uma base de dados de sequências, e identificar sequências que se assemelhem à sequência anteriormente consultada e que estejam acima de um certo grau de semelhança [1] [26].

Este *software* vai servir para efectuar alinhamentos simples de sequências, onde se pode perceber se uma determinada sequência de um organismo existe ou não em outros organismos, caso exista, total ou parcialmente, em que local desse mesmo organismo é que se encontra e quais as alterações encontradas.

4.1.4 Alinhamentos múltiplos

Um alinhamento múltiplo de sequências é um alinhamento, parcial ou completo, simultâneo de três ou mais sequências biológicas, geralmente proteínas, DNA ou RNA. De modo geral, assume-se que o conjunto de sequências de consulta que se coloca como entrada tem uma relação evolutiva pela qual compartilham uma linhagem e descendem de um ancestral comum. Do alinhamento resultante, pode-se inferir a homologia, e pode levar-se a cabo a análise filogenética para avaliar as origens evolutivas compartilhadas pelas sequências.

MUSCLE

MUSCLE [23] é um programa para criar alinhamentos múltiplos de aminoácidos ou sequências de nucleótidos. É fornecida uma gama de opções de modo a optimizar a precisão, velocidade, ou alguma relação entre os dois. Os alinhamentos múltiplos de sequências de proteínas são importantes em muitas aplicações, incluindo a estimativa da árvore filogenética, a previsão da estrutura e identificação do resíduo crítico. A maneira mais natural de formular o problema computacional é definir um modelo de evolução sequêncial que atribui probabilidades à edição da sequência elementar e procura um gráfico dirigido mais provável, na qual as extremidades representam as edições e os nós terminais representam as sequências observadas. Ainda não foi encontrado nenhum método que trate o tal gráfico. Uma alternativa heurística é encontrar um alinhamento múltiplo que optimiza a pontuação SP, ou seja, a pontuação somada do alinhamento de cada par da sequência [10] [9].

4.2 Arquitectura de Sistemas de Informação

Um sistema de informação (SI) é um sistema na qual o elemento principal é a informação. O objectivo principal de um SI é armazenar, tratar e fornecer informações de modo a apoiar as funções ou processos de uma organização. Claro que em SI não é apenas composto por um sistema automatizado (*software* e *hardware*) mas também por um sistema social em que envolve todas as pessoas, processos, informações e documentos. Assim sendo, o sistema automatizado vai interligar os elementos anteriores.

Existem então vários tipos de SI dependendo do tipo de questão que se coloca ou se pretende resolver. Ao desenvolver um SI é necessário ter em conta vários aspectos, o aspecto social e o ambiente real são alguns exemplos a ter em conta pois apesar do *software* estar a funcionar, pelo menos em ambiente de teste, em ambiente real e com o envolvimento social podem surgir questões do âmbito de utilização ou de falta de informação e o SI pode deixar de funcionar correctamente. É então necessário ter os objectivos, as informações a serem manipuladas, os processos e as pessoas que farão parte do SI.

4.2.1 Sistemas de bases de dados relacionais

Um sistema de gestão de bases de dados relacionais (SGBDR) tem uma BDR (base de dados relacional) que guarda as informações num conjunto de tabelas, cada uma contendo dados relativos a um determinado assunto. Uma base de dados relacional é uma colecção de vários dados organizados como um conjunto de tabelas formalmente descritas a partir do qual os dados podem ser acedidos facilmente e é criada usando o modelo relacional. O *software* utilizado numa base de dados relacional é chamado de SGBDR. Uma base de dados relacional é a escolha predominante no armazenamento de dados, pois a ma-

nutenção dos dados em tabelas relacionadas é muito eficiente uma vez que os dados só necessitam de ser arquivados uma vez o que reduz os requisitos de espaço em disco e torna mais fácil a actualização e a recepção dos mesmos. A linguagem padrão das Bases de Dados Relacionais é a *Structured Query Language*, ou (SQL) [5].

4.2.2 Arquitectura *web*

O propósito de utilizar uma arquitectura *web* é facilitar o acesso aos utilizadores permitindo que o mesmo manipule aplicações remotas usando apenas o *browser* como *interface*. Facilita a modificação e restruturação do sistema uma vez que este está alojado apenas num único local e a migração do sistema torna-se eficaz e rápida. É uma arquitectura que normalmente utiliza uma abordagem cliente-servidor existindo também a comunicação entre o servidor e bases de dados relacionais. Através de uma arquitectura destas é possível delinear todo o esqueleto de um sistema de informação *web* onde todas as demais partes se vão apoiar.

Arquitectura LAMP

A arquitectura usada foi a arquitectura LAMP (Linux, Apache HTTP Server, MySQL, PHP ou Python). É uma arquitectura baseada na comunicação entre cliente, servidor e base de dados, utilizando os protocolos de *internet* para essa mesma comunicação, tal como mostra a figura 4.1

O funcionamento da *web* baseia-se no modelo cliente-servidor, em que o utilizador requisita um ficheiro que se encontra num computador remoto através de um endereço (URL). Este pedido é normalmente efectuado através de um *web browser*. O servidor remoto compreende o pedido do cliente e devolve-lhe o ficheiro respectivo e quem está encarregue de atender esses pedidos nas arquitecturas LAMP é o Apache HTTP Server. O ficheiro requisitado normalmente vem codificado em HTML (*HyperText Markup Language*) de modo a que o *web browser* o possa compreender.

As descrições para as siglas LAMP são:

- Linux é um sistema operativo de código aberto baseado em Unix que é sistema seguro, estável e com um desempenho bastante agradável.

- Apache HTTP Server é um servidor *web* de código aberto que tanto pode ser usado em sistemas operativos Unix como Windows e é um servidor seguro, eficiente e extensível que fornece serviços HTTP em sincronia com os padrões actuais de HTTP [11].

- MySQL é a base de dados de código aberto mais conhecida actualmente, devido ao seu alto desempenho, alta confiabilidade e facilidade de uso [7]. Através do MySQL é então possível criar tabelas relacionais e armazenar dados nas mesmas

de modo a ser possível utilizar esses mesmos dados consoante a necessidade do utilizador.

- PHP (*Hypertext Preprocessor*) é uma linguagem de *script* de código aberto usada originalmente para o desenvolvimento de aplicações do lado do servidor, capaz de gerar conteúdo dinâmico na *web* e suporta também diferentes tipos de sistemas operativos [25]. É uma linguagem muito utilizada para a comunicação entre as bases de dados relacionais e o cliente *web*, sendo esta reconhecida no lado do servidor.

- Python é uma linguagem de programação de alto nível, multi paradigma, dinâmica, com extensas bibliotecas padrão e módulos para quase todas as necessidades. O Python tem um tipo de sistema dinâmico e de gerenciamento automático de memória como existe também em outras linguagens de programação. É frequentemente utilizado como uma linguagem de *script*, e utilizando algumas ferramentas já existentes é possível implementar esta linguagem de modo a executar *software* local.

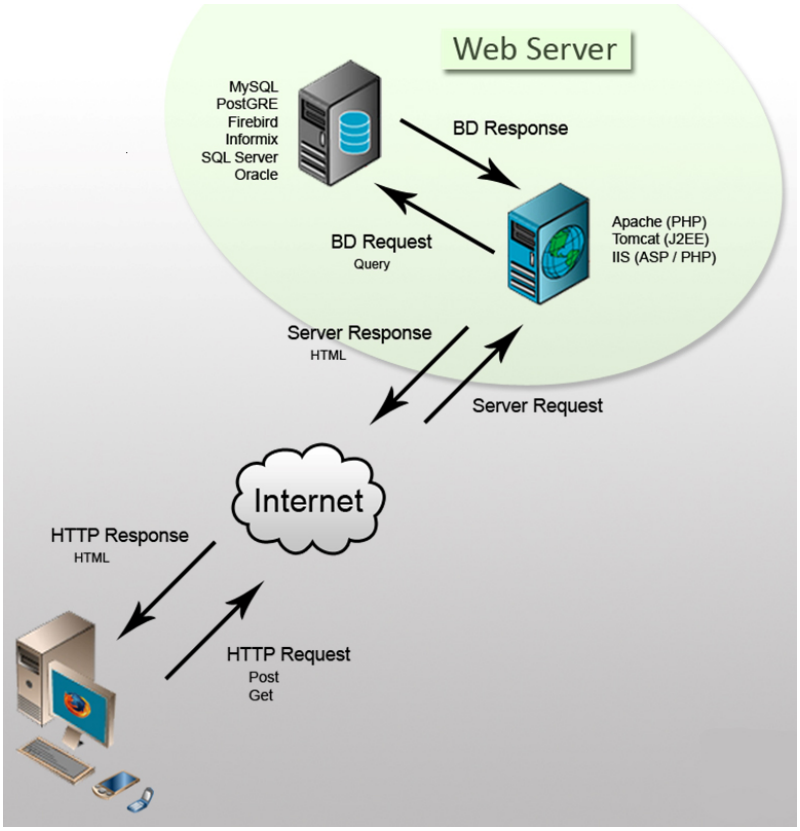


Figura 4.1: Arquitectura web.
(http://flavioaf.files.wordpress.com/2010/02/arquitetura_web.jpg)

4.2.3 Desenvolvimento web

Existem várias linguagens/formatos para desenvolvimento *web*. Seguidamente vamos explicar quais as linguagens que iremos usar na nossa aplicação e quais as suas utilidades, entre aquelas já acima referidas.

HTML

HTML5 (*HyperText Markup Language*) é uma linguagem de marcação utilizada para produzir páginas web de modo a que possam ser interpretados pelos *browsers*, 5 é a versão do HTML. Esta nova versão traz novas e importantes mudanças quanto ao papel do HTML no mundo da *web*, através destas novas funcionalidades como a semântica, acessibilidade e com novos recursos que só eram possíveis através da utilização de outras tecnologias [14].

O HTML5 adicionou novas funções, como as *tags* <*video*>, <*audio*>, <*header*> e elementos <*canvas*>, usados na nossa aplicação. Estas funções são projectadas para tornar mais fácil a inclusão e a manipulação de conteúdo gráfico e multimédia na *web* sem ter de recorrer a *plugins* e/ou APIs. Outros novos elementos, como <*section*>, <*article*>, <*header*> e <*nav*>, são projectados para enriquecer o conteúdo semântico dos documentos. Para usar esta versão do HTML, tal como as novas *tags*, é necessário que os *browsers* sejam compatíveis com a mesma, nos *browsers* mais antigos algumas das novas *tags* serão ignoradas. A *tag canvas* faz parte do HTML5 e o seu elemento permite criar uma área de desenho 2D, na qual se pode desenhar desde simples figuras geométricas, até imagens complexas. Têm apenas 2 atributos, o *width* e o *height*, que correspondem ao comprimento e altura da área de desenho respectivamente.

JavaScript

É uma linguagem de *script* para programação maioritariamente no lado do cliente de uma aplicação *web*. O principal uso desta linguagem é escrever funções para serem incluídas em páginas HTML, esta linguagem é definida com o uso das *tags* <*script*> ”código” <*/script*>. Deste modo é possível através das funções JavaScript ler e alterar conteúdo de elementos HTML, ler e mudar o estilo desses conteúdos que podem ser usadas para validar dados, como forma de validação de dados de entrada por parte do utilizador. Podem ser usadas para armazenar ou recuperar informações no computador do utilizador e também podem ser configuradas para serem executadas quando algo acontece, ou num determinado período especificado. Caso um *browser* não suporte a linguagem em questão é possível através das *tags* <*noscript*> ”mensagem” <*/noscript*> emitir uma mensagem ao utilizador a informar que o *browser* não suporta esta linguagem.

Ajax

É um acrónimo para *Asynchronous JavaScript and XML*, embora tenha XML no próprio nome não implica que este tenha obrigatoriamente de ser usado. Usando a tecnologia Ajax, uma determinada página HTML pode fazer chamadas de forma assíncrona para o servidor e carregar o conteúdo deste, que pode ser em formato XML, HTML, texto simples ou objectos JavaScript. Assim sendo, é um método usado de modo a que uma determinada página *web* seja alterada, recebendo ou não informação do servidor *web*, sem que seja necessário fazer *reload* à página em questão [6].

Alguns exemplos da interacção do Ajax.

- Validação de dados em tempo real, em que os dados que o utilizador introduz e que necessitam de ser validados, podem ser validados no lado do cliente antes de o utilizador enviar o formulário para o servidor com os seus dados.

- *Autocomplete*, à medida que o utilizador introduz caracteres sobre o que pretende é devolvida uma lista de resultados com as possibilidades de resposta.

- Carregar informação, com base num evento do utilizador, uma página HTML pode receber mais dados em segundo plano de modo a carregar informação extra ou novas paginas mais rapidamente.

- Controlo de interfaces e efeitos, controlos, tais como menus, tabelas de dados, editores de texto, calendários ou barras de progresso permitem uma melhor interacção do utilizador e a interacção com as páginas HTML, normalmente sem ser necessário que o utilizador recarregue a página.

- Actualizar dados e comunicar com o servidor, as páginas HTML podem consultar dados fornecidos por um servidor para actualizar dados da página, para isso um utilizador pode usar técnicas de Ajax para obter um conjunto de dados actuais sem recarregar a página inteira.

XML

eXtensible Markup Language é uma especificação que define uma sintaxe para a criação de documentos contendo dados de forma hierárquica, a grande vantagem da sua utilização é ser extensível, ou seja, todas as *tags* usadas podem ser criadas pelo autor com o nome que este entender. Outra das vantagens é a sua portabilidade, uma vez que este formato pode ser criado por uma aplicação, armazenando nesta os dados, e ser lido por outra aplicação diferente de modo a interpretar o seu conteúdo de forma rápida e eficaz.

CSS

Cascading Style Sheets é uma linguagem de estilo utilizada para definir a apresentação de documentos escritos em linguagens de marcação, como HTML ou XML. Umas das vantagens de usar a linguagem CSS é o facto de o tamanho do ficheiro CSS ser reduzido e com isto ajudar o carregamento de páginas *web* de forma rápida, e permitindo que as alterações feitas num local possam ser aplicadas a todo o documento, o que faz com que se ganhe tempo ao modificar estilos e formatações de documentos [17].

4.2.4 Processamento de informação genética

Existem várias ferramentas disponíveis para a realização do processamento de informação genética como as referidas no capítulo 4, no entanto é necessário haver maneira de comunicar com essas ferramentas e com a nossa aplicação. Neste caso usámos a linguagem Python com a inclusão do módulo Biopython para a comunicação entre o servidor e as ferramentas de processamento de informação genética.

Biopython

É um módulo com um conjunto de ferramentas que permitem fazer o processamento de informação na área da genética. Tem a capacidade de analisar arquivos de bioinformática em estruturas de dados utilizáveis em Python, esses arquivos podem ser iterados registo por registo ou indexados e acedidos através de uma interface ou dicionário. Através deste módulo é então possível executar algumas ferramentas externas, quer seja localmente ou via *web*, e deste modo trabalhar e interagir com os seus resultados. É possível realizar através das ferramentas disponíveis por este módulo várias operações, entre elas lidar com sequências e as suas características, lidar com alinhamentos simples ou múltiplos, lidar com bases de dados, entre outras operações.

Capítulo 5

Trabalho Realizado

O trabalho realizado tem como base a criação de um *Frontend* e um *Backend* de uma aplicação *web* de modo a que seja possível a visualização de mapas geonómicos e de alinhamentos simples e múltiplos. Uma das preocupações foi o facto de qualquer utilizador ou administrador da aplicação ser capaz de manusear com destreza a mesma não se perdendo no conteúdo e deste modo foram criados botões de ajuda em quase todas as opções e menus da aplicação. O nome escolhido para a nossa aplicação foi GIN [22], sendo este uma abreviatura para *Genome Inspector*.

5.1 Arquitectura da aplicação

Tal como mostra a figura 5.1 a nossa aplicação foi estruturada de maneira a que fosse possível existir comunicação entre as várias camadas existentes, ou seja, o cliente (utilizador) faz um pedido ao GIN e este é processado do lado do servidor que utiliza linguagens, ferramentas e tecnologias de pesquisa e processamento de informação de modo a obter o resultado pretendido e devolver o mesmo ao utilizador.

5.2 Design da interface

A função do *design* da interface é tornar a interacção do utilizador com a aplicação o mais simples e eficiente possível em termos da realização dos objectivos do mesmo. O processo de *design* deve equilibrar funcionalidades técnicas e elementos visuais de modo a criar um sistema que não é apenas operacional mas também útil e adaptável para alterar as necessidades do utilizador. Foi proposta uma abordagem ao *design* da aplicação como demonstra a figura 5.2, e foi aprovada por parte dos utilizadores que iriam trabalhar na nossa aplicação. Para ajudar na implementação do *design* da interface quer no *Backend* quer no *Frontend* foi usada uma colecção de ferramentas do Twitter Bootstrap de modo a embelezar a interface com o HTML e CSS próprio deles e de modo a tornar mais fácil a compatibilidade entre os *browsers* disponíveis nos vários sistemas operativos existentes.

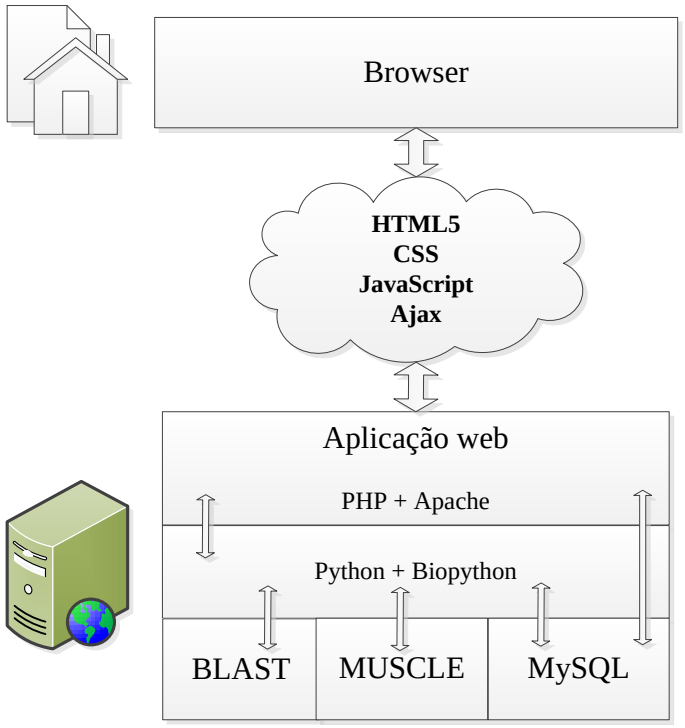


Figura 5.1: Arquitectura da aplicação.

5.3 Sistema de informação GIN

A estrutura da base de dados foi concebida em MySQL para permitir o acesso rápido, acomodar os dados heterogéneos e manter a integridade referencial dos mesmos. O conjunto de dados é constituído pelas propriedades dos organismos a estudar bem como as propriedades dos genes neles contidos. Foi desenvolvido um modelo relacional constituído por entidades que contêm os atributos que as caracterizam e pelas relações entre elas. Cada linha da entidade (tuplo) representa uma colecção de dados relacionados. Foi então criada uma base de dados (*gindb*) relacional com três tabelas (entidades) associadas, a tabela referente às Espécies, aos Organismos e aos Genes e uma tabela *Users* com as entradas dos administradores da aplicação. Na figura 5.3, mostramos as associações entre as entidades e quais os seus atributos.

Na entidade Espécies temos um *id* único para cada entrada, atribuído automaticamente pelo MySQL, e um nome referente ao nome da espécie. A entidade Organismos tem um *id* único, também este atribuído automaticamente, um nome do organismo, um *accession* (código do organismo para pesquisa do mesmo no GenBank), e duas referências para o *id* e para o nome das entradas correspondentes da tabela Espécies. A informação referente ao genoma completo de cada organismo foi guardada num ficheiro FASTA. Na entidade

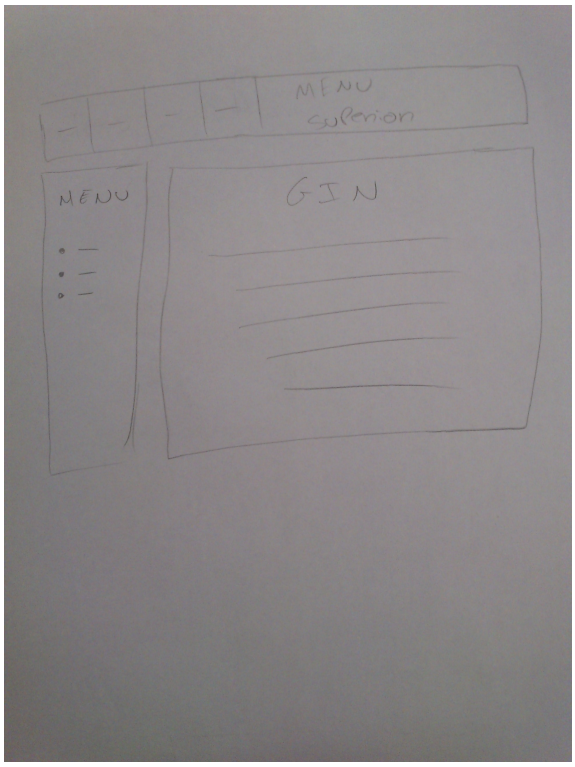


Figura 5.2: Protótipo de baixa fidelidade.

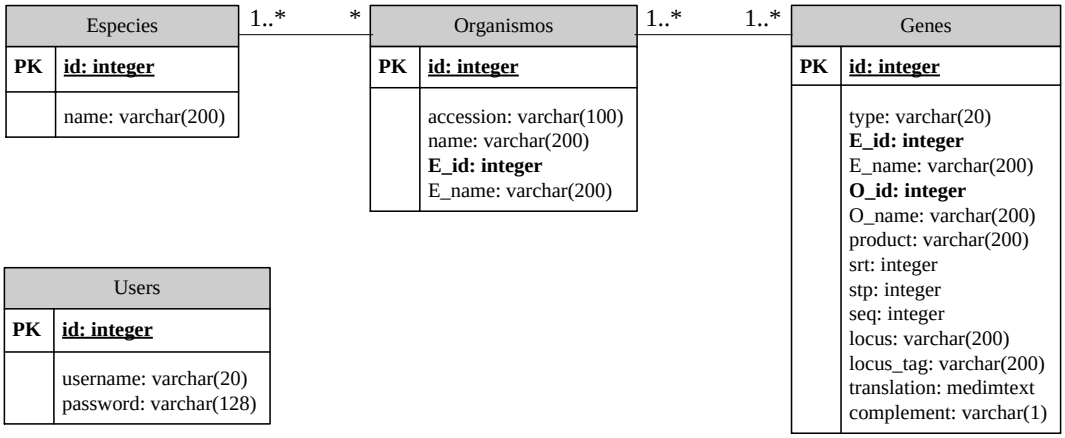


Figura 5.3: Estrutura da base de dados relacional.

Genes também temos um *id* único atribuído automaticamente para cada entrada, um *type* (tipo de gene em questão), *product* (o que o gene codifica, proteína ou RNA), *srt* (inicio da sequência do gene), *stp* (fim da sequência do gene), *seq* (a sequência em aminoácidos do gene), *locus* (nome do gene), *locus_tag* (a identificação do gene), *translation* (a translação da sequência do gene para proteina), *complement* (uma *flag* para identificar se

o gene está no sentido correcto ou se esta invertido) e as referências do *id* e do nome da tabela Organismos (*O_id, O_name*) e do *id* e do nome da tabela Espécies (E_id, E_name). Foi então guardada na entidade Genes as informações referentes a todos os genes encontrados em cada organismo. A entidade *Users* tem então o *username* do administrador e a *password* do mesmo.

Os atributos *accession* e *name* da entidade Organismos e os atributos *type*, *product*, *srt*, *stp*, *locus*, *locus_tag*, *translation* e *complement* da entidade Genes são preenchidos através de um ficheiro GenBank no formato XML. O atributo *seq* da entidade Genes é preenchido através do ficheiro FASTA que contém o genoma completo do organismo em questão. Tendo o *srt* e o *stp* do gene é possível ir ao genoma completo do organismo e retirar apenas a sequência pretendida desse gene. Os atributos *username* e *password* da entidade *Users* estão previamente inseridos na base de dados, servindo apenas de verificação para *login*. Os outros atributos são preenchidos conforme os dados introduzidos pelo administrador.

As entidades foram criadas com a opção *on delete cascade*, o que significa que ao eliminar uma entrada da entidade Espécies todas as entradas nas entidades Organismos e Genes referente a essa entrada também serão eliminadas. No caso de ser eliminada uma entrada na entidade Organismos serão também apagadas todas as entradas da entidade Genes referente a essa mesma entrada. Foram também criados índices de modo a melhorar o desempenho das perguntas à base de dados, testamos alguns índices e percebemos que estes seriam os mais eficazes nas respostas da base de dados. Os índices foram criados na entidade Genes e pensados nas *queries* mais frequentes à base de dados, foram criados então o *índex_E_name* (inclui um índice com todos os atributos *E_name*), *índex_O_name* (índice com todos os *O_name*), *índex_srt_stp* (índice com todos os *srt* e *stp*), *índex_type_product* (índice com todos os *type* e *product*), *índex_locus_tag* (índice com todos os *locus_tag*). Os índices *índex_srt_stp*, *índex_type_product* e *índex_locus_tag* são índices compostos, onde tem mais que um atributo atribuído ao mesmo índice.

5.4 *Backend*

As primeiras fases da nossa aplicação passaram pela criação do *Backend*, onde se pode dividir em duas partes, a gestão das bases de dados e a gestão dos organismos inseridos nas bases de dados. Vamos explicar em que contexto estão inseridos tanto a gestão das espécies como a gestão dos organismos nas secções seguintes, como está dividido todo o trabalho feito no *Backend* e como este funciona. Toda a informação do GIN está guardada em tabelas relacionais numa base de dados MySQL e em ficheiros com o formato FASTA.

Inicialmente estava previsto que a abordagem feita à aplicação fosse baseada no conceito de espécie bacteriana que continha organismos e que por sua vez esse mesmo organismo continha vários genes. Mais tarde verificou-se que o conceito espécie teria mais sentido se fosse mais abrangente e fosse possível ter quaisquer organismos de quaisquer

espécies na mesma tabela (entidade) e depois escolher quais os de interesse do utilizador, neste caso o conceito espécie indica o nome da base de dados de cada utilizador que poderá ter então os organismos das espécies que pretender. A entidade Espécie será mais tarde restruturada para um nome mais adequado e possivelmente com outros atributos.

5.4.1 Autenticação

Para aceder ao *backend* é necessário efectuar um *login* através da introdução de um *username* e de uma *password* como mostra a figura 5.4 A *password* como acima referido está guardada na entidade *Users* de forma encriptada. Depois dos dois campos estarem preenchidos são enviados para o lado do servidor onde este compara o *user* introduzido com o da base de dados e cifra a *password* inserida de modo a comparar as mesmas. Se forem as duas correctas então o *login* é feito com sucesso, caso contrário uma mensagem de erro é mostrada. Já dentro da página de *backend* é possível fazer *logout* e deste modo voltar à página anterior de *login*.

A autenticação apenas é necessária para o administrador do sistema e os utilizadores não necessitam de efectuar a autenticação para utilizar o GIN.

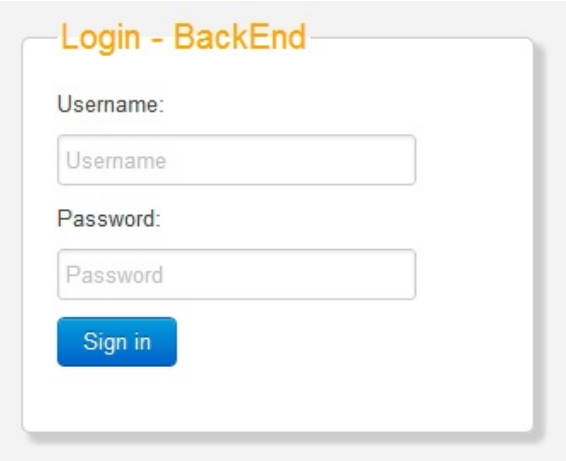


Figura 5.4: Login.
(http://xldb.fc.ul.pt/biotools/gin/)

5.4.2 Gestão da base de dados

A gestão das bases de dados dos utilizadores está organizada de modo a que seja possível criar novas entradas à entidade Espécies para cada utilizador, apagar essas mesmas entradas e todas as informações associadas a essa entrada e actualizar informação referente a cada entrada na entidade Espécies.

Criar novo ficheiro de base de dados

Tal como mostra a figura 5.5, o administrador apenas tem de inserir o nome (sem espaços) que o utilizador pretende para a sua base de dados. Ao introduzir o nome pretendido este é introduzido na entidade Espécies e é criado um ficheiro com o formato FASTA com esse nome onde mais tarde serão inseridos os genomas completos dos organismos dessa mesma base de dados.

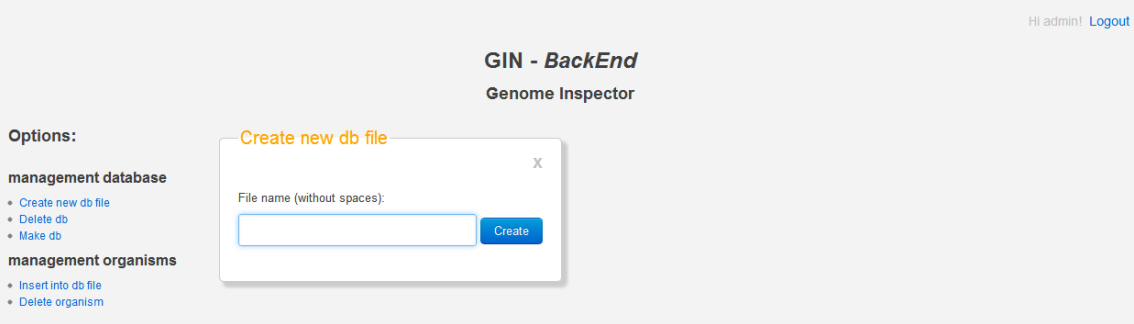


Figura 5.5: Criar nova base de dados.
(http://xldb.fc.ul.pt/biotools/gin/)

Apagar base de dados

Esta opção permite eliminar a entrada referente ao nome escolhido pelo administrador da entidade Espécies da base de dados MySQL e todos os organismos e genes associados a essa mesma entrada. Elimina também o ficheiro FASTA criado anteriormente com o nome igual escolhido pelo administrador e todo o seu conteúdo. A figura 5.6, mostra a opção de apagar uma base de dados de um utilizador.

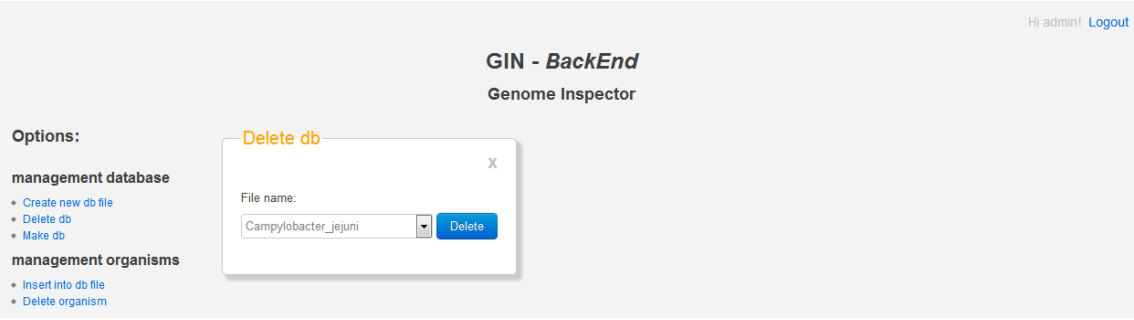


Figura 5.6: Apagar nova base de dados.
(http://xldb.fc.ul.pt/biotools/gin/)

Actualizar base de dados

Após a criação da base de dados e de serem introduzidos os organismos pretendidos é preciso ter alguma forma de transformar o ficheiro FASTA, com o nome que o administrador introduziu e com os genomas completos dentro desse mesmo ficheiro separados pelo símbolo > seguido do nome ou descrição desses organismos, numa base de dados BLAST. Para essa transformação é necessário executar o comando *makeblastdb* [4], de modo a que o BLAST possa identificar todas as sequências existente no ficheiro FASTA e deste modo ser possível ao BLAST executar alinhamentos e comparações de sequências através desse mesmo ficheiro. O comando *makeblastdb* tem de ser executado sempre que for introduzido um novo organismo num determinado ficheiro FASTA de modo a que o BLAST reconheça que existem novos organismos nesse ficheiro. Este comando é executado no lado do servidor e através do módulo Biopython. A figura 5.7, mostra então como se pode actualizar as informações dos ficheiros FASTA de modo a que o BLAST as reconheça.

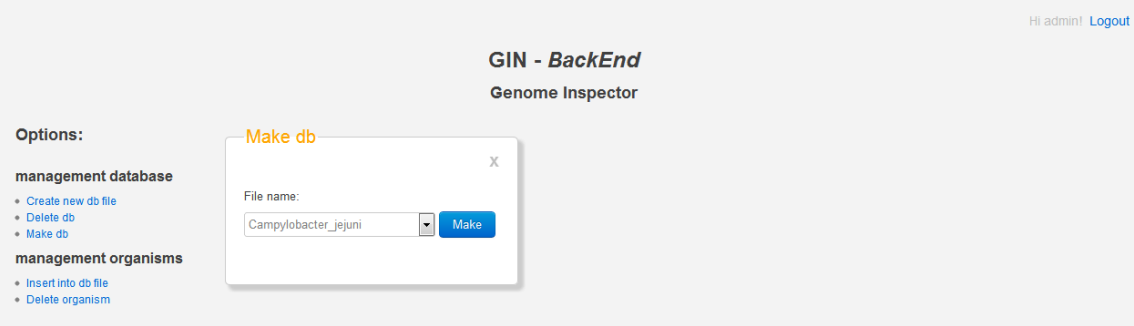


Figura 5.7: Actualizar base de dados.
(http://xldb.fc.ul.pt/biotools/gin/)

5.4.3 Gestão dos Organismos

A gestão dos organismos permite a introdução e remoção de organismos tanto na base de dados MySQL como nos ficheiros FASTA existentes. Está então dividida em duas opções, a opção de inserir organismos e a opção de apagar organismos como iremos explicar de seguida.

Inserir organismos

Nesta opção tal como mostra a figura 5.8, é necessário escolher em qual a base de dados em que se quer inserir determinados organismos e depois escolher o modo de como se quer inserir os organismos, ou através de códigos GenBank (num máximo ate 10 de

cada vez separados por vírgula) ou através de um ficheiro externo com os organismos em formato GenBank.

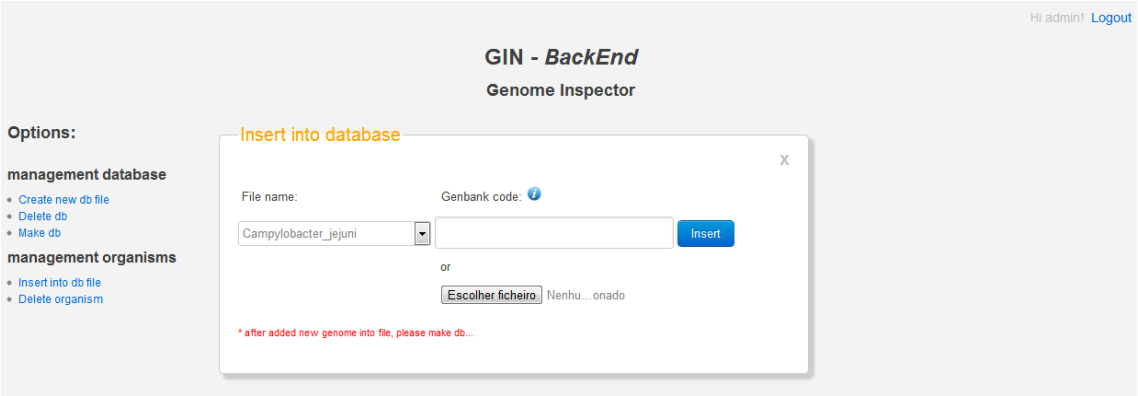


Figura 5.8: Inserir organismos na base de dados.
(http://xldb.fc.ul.pt/biotools/gin/)

Caso o modo de inserção escolhida seja através de códigos GenBank estes códigos são tratados um a um do lado do servidor. Primeiramente é verificado se esses códigos existem na base de dados MySQL, os códigos que ainda não estão nessa base de dados são pedidos aos serviços *web* do GenBank (através do Biopython) de modo a obtermos os ficheiros desses códigos nesse mesmo formato e no formato XML, caso esses códigos não existam uma mensagem de erro é mostrada. Após termos esses dois ficheiros, convertemos o ficheiro com o formato GenBank num novo ficheiro com o formato FASTA onde contém o genoma completo desse organismo. Esse ficheiro é lido e o seu conteúdo é anexado ao ficheiro escolhido inicialmente pelo administrador através do nome da base de dados. O ficheiro com o formato XML é lido e são introduzidos todos os campos encontrados referentes à entidade Organismos e à entidade Genes do organismo em questão e introduzidos nas entidades respectivas da base de dados MySQL. Modelo de actividades desta opção na figura 2.4.

Caso a escolha seja feita através do carregamento de um ficheiro externo (em formato GenBank) este é lido e enviado o conteúdo para o lado do servidor. Esse conteúdo é escrito num novo ficheiro temporário e é retirado o código GenBank e verificado se esse código existe na base de dados MySQL, caso não exista é feita uma conversão do ficheiro num novo ficheiro temporário com o formato FASTA onde contém o genoma completo desse organismo. Esse ficheiro é lido e o seu conteúdo é anexado ao ficheiro escolhido inicialmente pelo administrador através do nome da base de dados. Depois é feito um *parse* do ficheiro com o formato GenBank com o Biopython e os campos encontrados referentes aos genes do organismo são inseridos na entidade Genes da base de dados MySQL. Modelo de actividades relativo a esta opção de inserção na figura 2.5.

Após a inserção dos organismos quer seja através dos códigos GenBank quer seja

através do carregamento de um ficheiro externo com genoma no formato GenBank é necessário efectuar a opção *make db* como referido acima. Este comando serve para actualizar a base de dados FASTA do utilizador.

Apagar organismos

Para apagar um organismo é necessário, tal como mostra a figura 5.9, escolher uma base de dados e um organismo nela inserido.

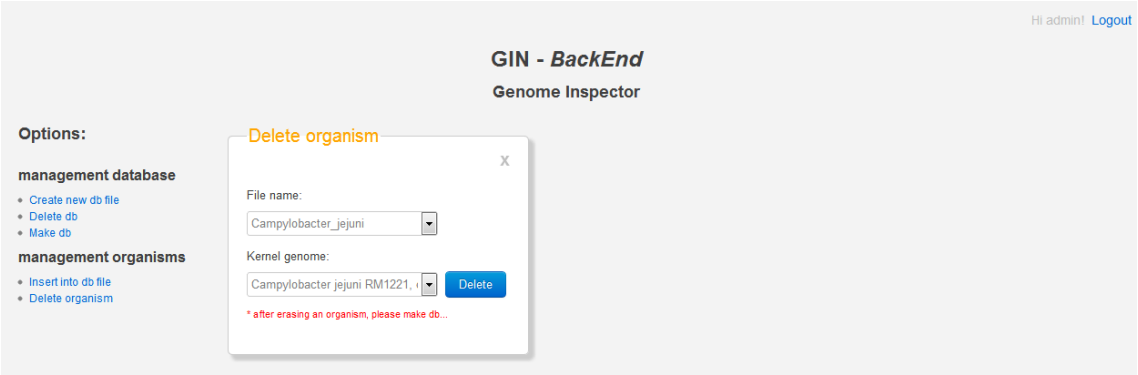


Figura 5.9: Apagar organismos na base de dados.
(http://xldb.fc.ul.pt/biotools/gin/)

Após escolhido o nome da base de dados e do organismo em questão essa informação é enviada para o lado do servidor e é eliminada a entrada da entidade Organismos e as entradas correspondentes da entidade Genes na base de dados MySQL e é também eliminada toda a informação referente ao organismo (genoma completo e descrição) no ficheiro FASTA correspondente ao nome da base de dados. Uma vez apagadas as informações relativas ao organismo será também necessário efectuar a opção *make db* para actualização da base de dados FASTA.

5.5 *Frontend*

O nosso *Frontend* foi construído a pensar nas tecnologias *web* existentes e de modo a que qualquer utilizador tivesse facilidade em navegar pela nossa aplicação e deste modo foi construída uma *interface* minimalista com cores contrastantes tal como mostra a figura 5.10.

A nível de *design* do *Frontend* pretendemos ser o mais fiel possível ao protótipo de baixa fidelidade apresentado em 5.2. No menu lateral temos três opções de pesquisa incluídas. A pesquisa por gene, onde se pode efectuar uma pesquisa através de um determinado gene guardado na nossa base de dados relacional. Pesquisa por região, onde

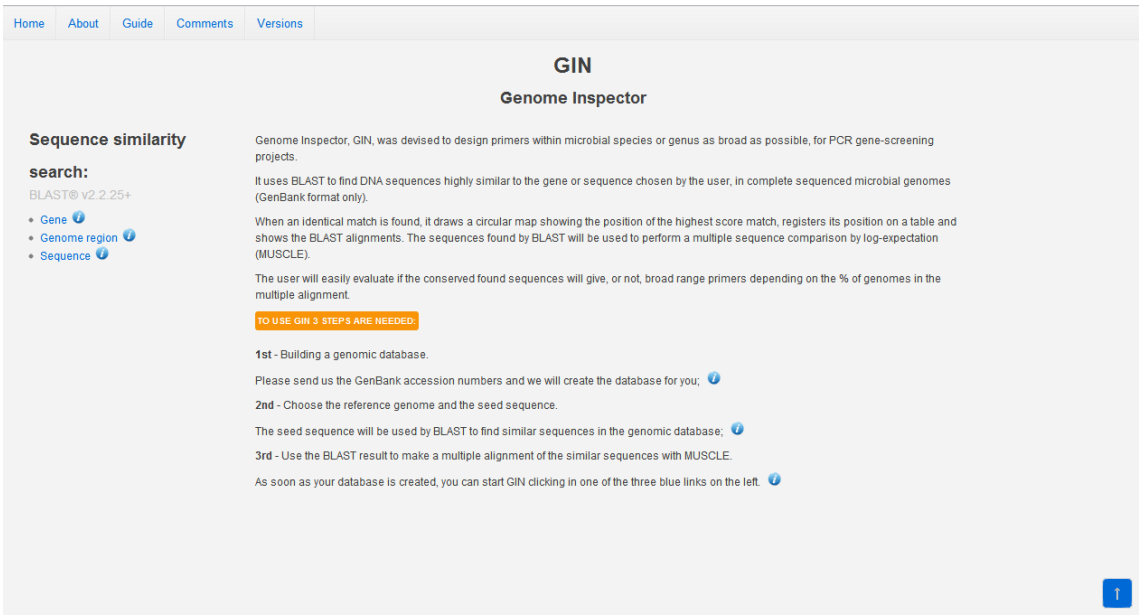


Figura 5.10: Frontend.
(http://xldb.fc.ul.pt/biotools/gin/)

é possível através de um valor de início e fim de uma sequência pretendida de um organismo, saber qual é essa sequência e fazer a pesquisa. Pesquisa por sequência, onde é dada a possibilidade ao utilizador de escolher entre inserir directamente a sua sequência manualmente no GIN ou inserir através de um ficheiro externo. As três opções de pesquisa acima referidas têm modos de vista iguais, sendo eles, o alinhamento simples de sequências, utilizando o sofware BLAST para o efeito, Gráficos circulares, onde foi usada a tecnologia HTML5 com o elemento *canvas* para o desenho dos mesmos e com a possibilidade de fazer *download* de cada gráfico apresentado, Gráficos lineares, também com recurso à tecnologia *canvas* e alinhamentos múltiplos, utilizando o MUSCLE e sendo possível devolver o resultado do mesmo em vários formatos, HTML, ClustalW ou Fasta. Temos também um menu superior onde se pode saber sobre este projecto e de todos os intervenientes no mesmo. Temos então as opções de *Home* (página principal), *About* (acerca do projecto e dos intervenientes), *Guide* (manual de utilização da aplicação), *Comments* (enviar mails ao administrador do projecto) e *Versions* (versões de cada *software* usado). Por fim temos a parte central onde é colocada a descrição da nossa aplicação bem como o preenchimento das opções das pesquisas e os seus resultados nos diferentes modos de vista.

Primeiramente vamos descrever as diferentes opções de pesquisa que disponibilizamos na nossa aplicação e quais os resultados provenientes das mesmas.

5.5.1 Pesquisa por gene

Na pesquisa por gene tal como mostra a figura 5.11, temos de escolher alguns campos de preenchimento.

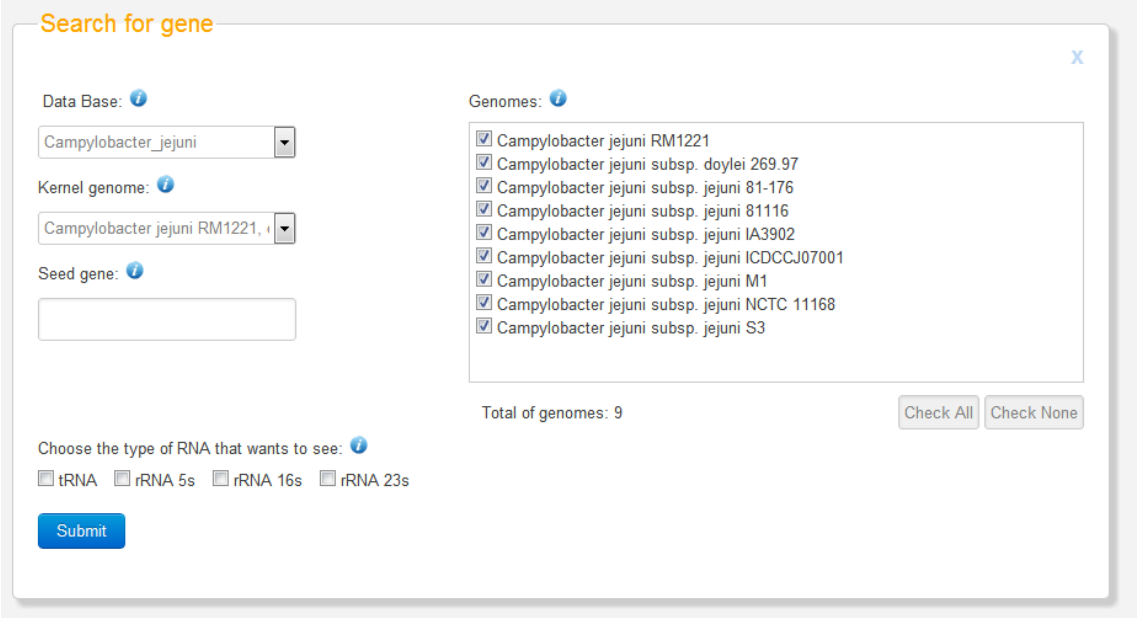


Figura 5.11: Pesquisa por gene.
(http://xldb.fc.ul.pt/biotools/gin/)

O utilizador deverá escolher qual a base de dados que pretende na opção *Data Base*, neste momento será mostrado na opção *Kernel genome* e na opção *genomes* todos os genomas correspondentes a essa base de dados escolhida. Após escolher a base de dados escolhe-se um genoma na opção *Kernel genome* para "correr" contra os genomas escolhidos pelo utilizador na opção *Genomes*. Escolhe-se então quais os genomas pretendidos na opção *Genomes* para efectuar o alinhamento de sequências através do BLAST.

De seguida escolhe-se um gene na opção *seed gene* relativo ao organismo seleccionado na opção *Kernel genome* e para isso basta escrever uma letra/número correspondente ao *locus_tag* desse mesmo gene que as opções dos genes existentes serão mostradas e filtradas à medida que se introduz mais letras/números. Temos uma função *autocomplete* que vai fazendo essa mesma filtragem à medida que se introduz os caracteres na opção *Seed gene*, como mostra a figura 5.12, através de pedidos consecutivos à base de dados relacional tentando perceber se existe algum gene daquele organismo onde o *locus_tag* tenha esse caractere contido. Caso o caractere introduzido pelo utilizador não exista, nenhuma lista de opções é mostrada.

Caso o utilizador queira visualizar no modo de vista de gráficos circulares os tipos de RNA disponíveis basta escolher quais os que quer ver e estes vão serão desenhados nos gráficos circulares representados por linhas com cores diferentes para cada tipo de RNA

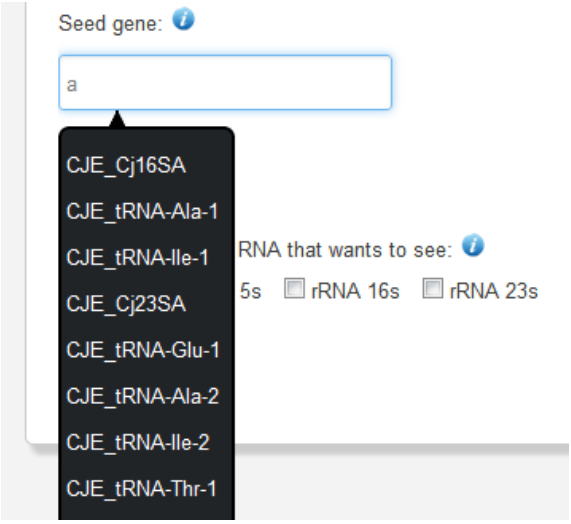


Figura 5.12: Exemplo de autocomplete.
(http://xldb.fc.ul.pt/biotools/gin/)

escolhido.

Após os campos estarem todos preenchidos conforme as preferências do utilizador, basta carregar no botão *Submit* de forma a processar toda a informação escolhida. Essa informação é enviada para o lado do servidor e é obtida a sequência do gene escolhido através de uma pergunta à base de dados MySQL com o nome da base de dados escolhida, do organismo e do gene. Tendo a sequência do gene escrevemos essa sequência para um ficheiro temporário que será usado com o BLAST. É executado o BLAST sobre o ficheiro criado anteriormente com a sequência e sobre a base de dados FASTA como nome da mesma escolhida inicialmente. O resultado é dado num ficheiro com o formato XML. O resultado das opções de vista é obtido a partir desse ficheiro XML através do alinhamento do BLAST. Temos então um pequeno exemplo de um resultado de um alinhamento simples de sequências efectuado pelo BLAST em formato XML na figura 5.13

Assim que o BLAST for executado e as várias opções de vista estiverem completas serão mostradas as opções de vista da nossa aplicação como podemos ver na figura 5.14 Para cada opção destas é mostrado um resultado diferente como vamos explicar mais abaixo.

Com base no resultado obtido do alinhamento do BLAST em 5.13 é criada uma tabela para cada modo de vista com os dados de cada organismo obtido do alinhamento feito, tal como mostra a figura 5.15, da pesquisa por gene. É possível escolher quais os resultados que se quer ver tendo ou não seleccionada a *checkbox* de cada organismo da tabela. Se clicarmos no nome do organismo a página é direccionada para o resultado referente ao organismo em questão.

```
  <Hit_num>1</Hit_num>
  <Hit_id>CP001960.1</Hit_id>
  <Hit_def>Campylobacter jejuni subsp. jejuni S3, complete genome.</Hit_def>
  <Hit_accession>CP001960.1</Hit_accession>
  <Hit_len>1681364</Hit_len>
- <Hit_hsps>
  - <Hsp>
      <Hsp_num>1</Hsp_num>
      <Hsp_bit-score>2795.10121822092</Hsp_bit-score>
      <Hsp_score>1513</Hsp_score>
      <Hsp_evalue>0</Hsp_evalue>
      <Hsp_query-from>1</Hsp_query-from>
      <Hsp_query-to>1513</Hsp_query-to>
      <Hsp_hit-from>37394</Hsp_hit-from>
      <Hsp_hit-to>38906</Hsp_hit-to>
      <Hsp_query-frame>1</Hsp_query-frame>
      <Hsp_hit-frame>1</Hsp_hit-frame>
      <Hsp_identity>1513</Hsp_identity>
      <Hsp_positive>1513</Hsp_positive>
      <Hsp_gaps>0</Hsp_gaps>
      <Hsp_align-len>1513</Hsp_align-len>
      <Hsp_qseq>TTTTTATGGAGAGTTTGATCCTGGCTCAGAGTGAACGCTGGCGGCGTGCCTAATACATGCAAGTCGAACGATGAAGCTTCTAGCTTGCTAGAAGTG
      <Hsp_hseq>TTTTTATGGAGAGTTTGATCCTGGCTCAGAGTGAACGCTGGCGGCGTGCCTAATACATGCAAGTCGAACGATGAAGCTTCTAGCTTGCTAGAAGTG
      <Hsp_midline>||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
    </Hsp>
```


Figura 5.13: Resultado do BLAST em formato XML.

Results view options

- Circular graphic
- Sequence alignment
- Linear graphic
- Multiple alignment

Figura 5.14: Opções de vista.
(http://xldb.fc.ul.pt/biotools/gin/)

#	Nº	Name	Score	Evalue	Start	End
☑	1	Campylobacter jejuni subsp. jejuni S3	1513	0	37394	38906
☑	2	Campylobacter jejuni RM1221	1513	0	37396	38908
☑	3	Campylobacter jejuni subsp. jejuni IA3902	1507	0	39260	40772
☑	4	Campylobacter jejuni subsp. jejuni NCTC 11168	1504	0	39249	40761
☑	5	Campylobacter jejuni subsp. doylei 269.97	1504	0	39388	40900
☑	6	Campylobacter jejuni subsp. jejuni 81116	1504	0	37565	39077
☑	7	Campylobacter jejuni subsp. jejuni ICDCCJ07001	1501	0	37381	38893
☑	8	Campylobacter jejuni subsp. jejuni M1	1498	0	37543	39055
☑	9	Campylobacter jejuni subsp. jejuni 81-176	1498	0	39155	40667

Figura 5.15: Tabela dos resultados da pesquisa por gene.
(http://xldb.fc.ul.pt/biotools/gin/)

Alinhamento Simples

Os alinhamentos simples são feitos tendo em conta os primeiros blocos de resultados de cada organismo encontrado no ficheiro XML resultante do BLAST em 5.13. É apresentada essa informação em formato texto e é apresentado o alinhamento proveniente do resultado obtido. Para a realização dos alinhamentos simples com base nos dados do ficheiro XML a informação que necessitamos é o nome de cada organismo (*Hit_def*), o código GenBank (*Hit_accession*), o tamanho (*Hit_len*) do genoma completo desse organismo, o *Bit score* (*Hsp_bit-score*), o *score* (*Hsp_score*), o *evalue* (*Hsp_evalue*), o início (*Hsp_hit-from*) e o fim (*Hsp_hit-to*) de cada sequência desse organismo, o *identity* (*Hsp_identity*) e o *gaps* (*Hsp_gaps*). Assim podemos indicar essas mesmas informações captadas referentes a cada organismo tal como mostra a figura 5.16.

```
Name: Campylobacter jejuni subsp. jejuni S3, complete genome.
Acession: CP001960.1
Length: 1681364
Bit score: 2795.10121822092
Score: 1513
Evalue: 0
Hit from: 37394
Hit to: 38906
Identity: 1513
Gaps: 0
```

Figura 5.16: Dados do alinhamento simples de sequências.
(http://xldb.fc.ul.pt/biotools/gin/)

Pretende-se mostrar o alinhamento efectuado pelo BLAST, no entanto o modo de vista desse alinhamento foi alterado de maneira a ser mais perceptível às alterações encontradas nas sequências resultantes do BLAST. Em consequência resultam três linhas de alinhamentos em que a primeira linha corresponde à sequência do organismo escolhido pelo utilizador na opção *Kernel genome* e a terceira linha corresponde aos genes que se escolheu para comparar na opção *Genomes*. Sobre a primeira linha está representado também o bloco numérico da sequência do organismo em questão e sob a terceira linha é apresentado o bloco numérico de cada organismo encontrado. Estes passos acima referidos são repetidos enquanto houver blocos de resultados no ficheiro XML proveniente do BLAST. O resultado é alterado de maneira a que cada caractere da primeira linha seja comparado com o da terceira, caso sejam iguais e colocado um . na primeira e terceira linha e a letra na segunda linha. Se houver alterações ao alinhamento então o caractere da primeira linha é colocado nessa mesma linha. O caractere da terceira linha é colocado na mesma linha e o caractere da segunda passa a ter um espaço. Para melhor percepção vamos mostrar um resultado do alinhamento simples como o da figura 5.17

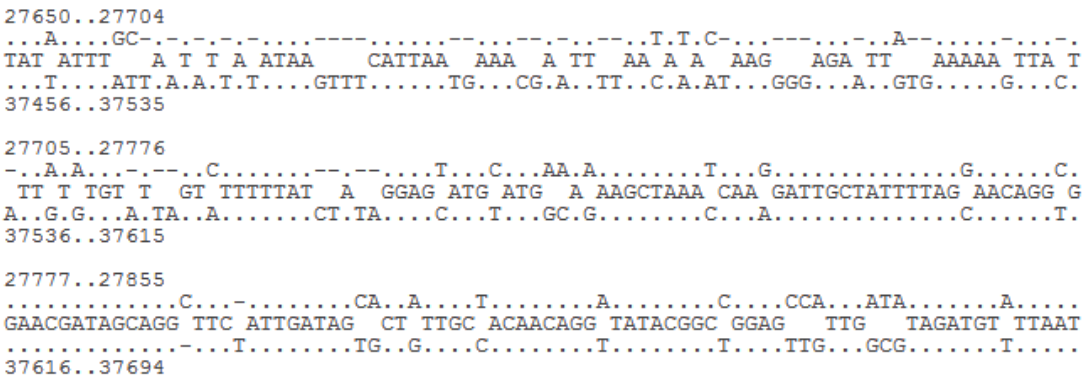


Figura 5.17: Resultado do alinhamento simples de sequências.
(http://xldb.fc.ul.pt/biotools/gin/)

Gráficos Circulares

Tal como os alinhamentos simples também os gráficos circulares são construídos com base nos dados do ficheiro XML resultante do BLAST em 5.13. Estes gráficos têm como objectivo apresentar a posição exacta da sequência encontrada. Essa sequência é apresentada na forma de uma figura tipo triângulo e num determinado organismo sob a forma de um círculo. São necessárias algumas etapas para realizar a apresentação dos gráficos circulares. Com base nos dados do ficheiro XML necessitamos de informações relativas ao organismo em questão e à sequência encontrada.

O desenho dos gráficos é feito no lado do cliente através do elemento *canvas* com HTML5 e JavaScript e é desenhado um círculo de cor azul onde o tamanho do mesmo vai corresponder ao comprimento do organismo, ou seja, sabemos que o círculo começa sempre em 1 e acaba com o tamanho igual ao do *Hit_len*. Tendo o início e o fim da sequência desenham-se duas linhas na fronteira do círculo, uma para o início e outra para o fim com um comprimento definido e igual para todos os gráficos. Depois de ter as duas linhas desenha-se um arco que una as extremidades das mesmas de modo a obter uma figura como a da 5.18. Caso o início da sequência seja maior que o final, a sequência está invertida e então o círculo será pintado de cor igual à da figura que representa a sequência. Verifica-se se existem tipos de RNA escolhidos de modo a serem representados por linhas de cores diferentes por cada tipo de RNA nos gráficos circulares já desenhados anteriormente. No caso da figura 5.18, nenhum tipo de RNA foi escolhido. Existe também a opção de fazer *download* da imagem de cada gráfico circular, para isso basta clicar no gráfico que se pretende fazer *download*. Para esta opção foi necessário converter toda a área de desenho *canvas* de cada gráfico circular para uma imagem de modo a que fosse possível ao utilizador efectuar o *download*. Todos os passos acima referidos são repetidos enquanto houver blocos de resultados por parte do ficheiro XML que obtivemos através do BLAST.

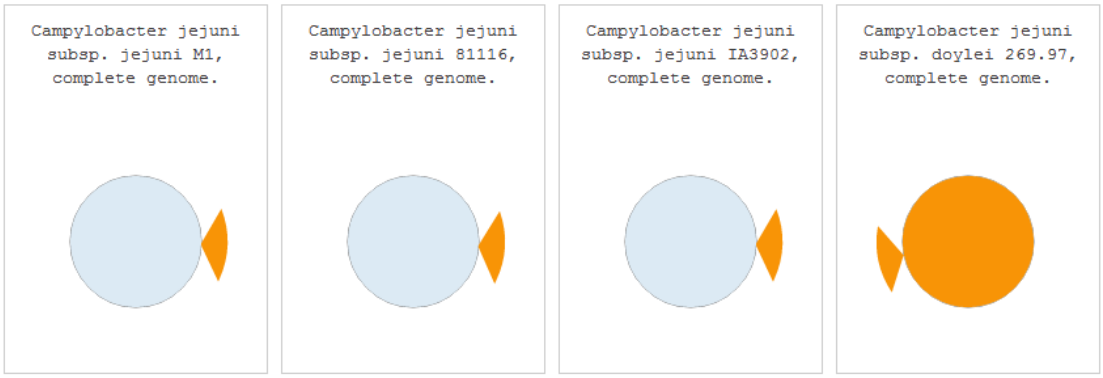


Figura 5.18: Gráficos circulares da pesquisa por gene.
(http://xldb.fc.ul.pt/biotools/gin/)

Gráficos Lineares

Nesta opção o objectivo é mostrar em forma de seta todos os genes encontrados numa determinada sequência. O sentido da seta indica se determinado gene se encontra em complemento ou não, ou seja, se o início do gene for menor que o final a seta é desenhada da esquerda para a direita, caso contrário, é desenhada da direita para a esquerda. Esta opção não está disponível na pesquisa por gene pois não tinha sentido mostrar apenas uma seta, uma vez que estamos na pesquisa por gene e cada seta representa um gene. Para se saber se este está ou não em complemento podemos ver nos gráficos circulares de acordo com a cor dos mesmos.

Alinhamentos Múltiplos

Nesta opção pretende-se mostrar alinhamentos múltiplos em vários tipos de formatos através do *software* MUSCLE e esses alinhamentos são feitos com base no resultado proveniente do BLAST. Os dados que necessitamos para a realização dos alinhamentos múltiplos são os nomes dos organismos (*Hit_def*), e o início *(Hsp_hit-from*) e o fim (*Hsp_hit-to*) de cada sequência encontrada e a opção de saída do resultado do alinhamento. Para a realização dos alinhamentos múltiplos primeiramente é criada uma tabela como a da figura 5.19, onde são colocados os nomes, o início e o fim de cada sequência encontrada no resultado do BLAST. É possível editar cada campo da tabela referente ao início e ao fim de cada sequência ou deixar como está, campos pré preenchidos pelo resultado do BLAST em 5.13. Após o preenchimento é necessário escolher o formato de saída do alinhamento múltiplo que o utilizador pretende visualizar, podendo ser em HTML, FASTA ou ClustalW.

Para finalizar basta fazer *submit*, carregar no botão *Muscle it*. Ao submeter os dados estes serão verificados de modo a perceber quais são os dados que se pretende fazer

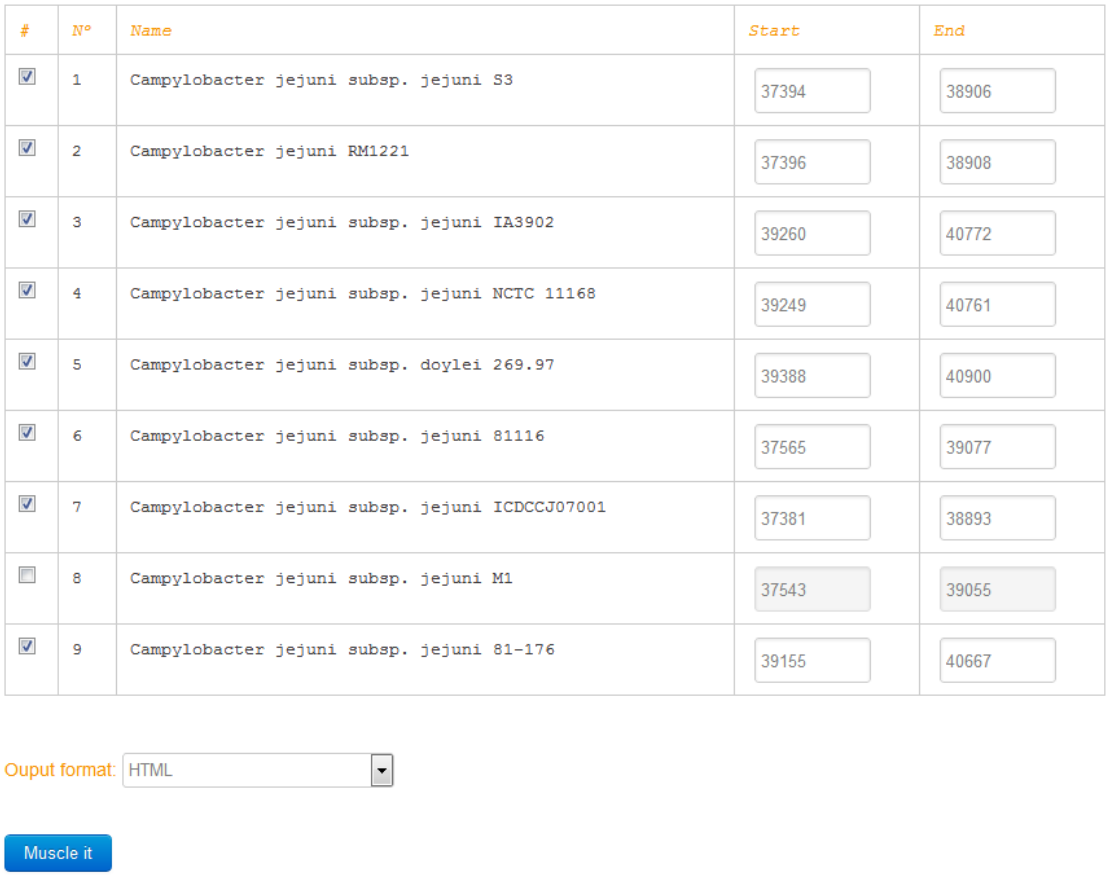


Figura 5.19: Tabela de alinhamentos múltiplos.
(http://xldb.fc.ul.pt/biotools/gin/)

alinhamento múltiplo, uma vez que só os campos com a *checkbox* activa é que serão contabilizados, e se esses dados estão no formato correcto, formato numérico. Antes da realização do alinhamento múltiplo através do MUSCLE é mostrado um aviso com a informação aproximada do tempo que o MUSCLE irá demorar até obter o resultado pretendido. Estes tempos foram obtidos através de vários pedidos com várias sequências e de tamanhos diferentes. Caso o pedido seja demasiado grande, intervalos de início e fim de todas as sequências, um aviso será mostrado de modo a diminuir os intervalos para que seja possível efectuar o alinhamento.

Na base de dados FASTA, com o nome escolhido inicialmente na pesquisa por gene, obteremos todas as sequências escolhidas pelo utilizador através do início e do fim da mesma relativamente a cada organismo seleccionado e criar um ficheiro FASTA temporário. A informação desse ficheiro contém o número de cada organismo referenciado na tabela e a sua sequência de modo a que seja lida pelo MUSCLE. É então executado o *software* MUSCLE com o ficheiro temporário já criado e com a opção do formato de saída do resultado igual ao escolhido pelo utilizador. De 5 em 5 segundos é verificado se o re-

sultado do MUSCLE já existe e durante um período limitado de tempo (10 minutos), caso esse período expire é pedido para o utilizador rever os dados e voltar a fazer o pedido. Quando o resultado estiver completo verificamos se o mesmo devolvido pelo MUSCLE tem um tamanho adequado para ser integrado na aplicação, até 1.5 MB. Caso o tamanho do resultado passe esse valor um aviso é mostrado de modo a que o utilizador possa ver esse mesmo resultado numa nova página *web*.

Podemos ver então um resultado de um alinhamento múltiplo relativo a uma pesquisa por gene como o da figura 5.20

Figura 5.20: Resultado do MUSCLE da pesquisa por gene.
(http://xldb.fc.ul.pt/biotools/gin/)

5.5.2 Pesquisa por região

Na pesquisa por região seguimos o mesmo raciocínio da pesquisa por gene, no entanto em vez a pesquisa ser feita através de um determinado gene existente é feita através de uma região de um determinado organismo. Tal como mostra a figura 5.21, é necessário preencher alguns campos para efectuar a pesquisa por uma determinada região de um organismo.

Para a realização da pesquisa por região é necessário escolher qual a base de dados que se pretende na opção *Data Base* e neste momento será mostrado na opção *Kernel genome* e na opção *Genomes* todos os genomas correspondentes a essa base de dados escolhida. Após escolher a base de dados selecciona-se um genoma na opção *Kernel genome* para "correr" contra os genomas escolhidos pelo utilizador na opção *Genomes*.

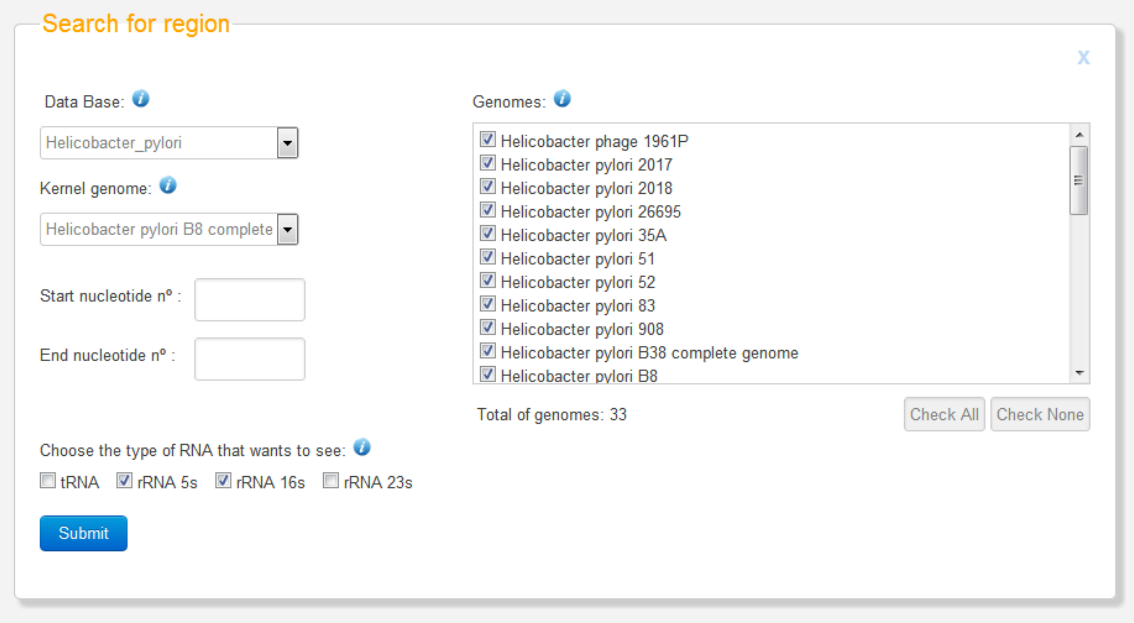


Figura 5.21: Pesquisa por região.
(http://xldb.fc.ul.pt/biotools/gin/)

Escolhe-se quais os genomas pretendidos na opção *Genomes* para efectuar o alinhamento de sequências através do BLAST. De seguida temos de indicar qual o valor de início da sequência pretendida relativa ao organismo na opção *Start nucleotide nº* e indicar qual o final da sequência na opção *End nucleotide nº*. O número de *Start* tem de ser inferior ao de *End*. Escolher ou não algum/alguns tipos de RNA que se pretende ver nos gráficos circulares e por fim submeter os dados através do botão *Submit*.

Após os dados serem submetidos são enviados para o servidor e deste modo proceder à recolha de informação necessária para executar o BLAST e obteremos o seu resultado em XML, como em 5.13, para o pudermos trabalhar. Os passos necessários até obtermos o resultado do BLAST em formato XML passam por, obter a sequência pretendida através da pesquisa à base de dados FASTA com o nome da opção *Data base* escolhida. O ficheiro FASTA tem todos os genomas completos de cada organismo incluídos nessa base de dados, assim sendo, vamos encontrar e guardar todo o genoma completo do organismo escolhido. Tendo então o organismo completo guardado e sabendo que a primeira letra do genoma completo de qualquer organismo é como se fosse o nosso número 1, basta escolher a sequência pretendida através do início e fim da mesma. Tendo a sequência pretendida, escrevemos essa sequência para um ficheiro temporário que será usado com o BLAST.

É executado então o BLAST sobre o ficheiro criado anteriormente com a sequência e sobre a base de dados FASTA como nome da base de dados escolhida inicialmente. O resultado é dado num ficheiro com o formato XML.O resultado das opções de vista

é obtido a partir desse ficheiro XML através do alinhamento do BLAST. Tendo então o resultado do BLAST relativo ao alinhamento da sequência que introduzimos é criada uma tabela semelhante à figura 5.15, como podemos ver na figura 5.22

#	Nº	Name	Score	Evalue	Start	End
☑	1	Helicobacter pylori B8	4001	0	4000	8000
☑	2	Helicobacter pylori P12	3497	0	1590224	1586218
☑	3	Helicobacter pylori India7	3415	0	1500991	1504984
☑	4	Helicobacter pylori 52	3410	0	1482815	1478890
☑	5	Helicobacter pylori B38 complete genome	3366	0	1486164	1482417

Figura 5.22: Tabela de resultados da pesquisa por região.
(http://xldb.fc.ul.pt/biotools/gin/)

Alinhamento Simples

Processa-se da mesma maneira que na pesquisa por gene tal como explicado em cima. Obtendo o resultado do BLAST em formato XML em 5.13, verifica-se o primeiro bloco de resultados de cada organismo e extrai-se a informação necessária para mostrar o alinhamento tal como mostra a figura 5.16 e a figura 5.17

Gráficos Circulares

O processamento também é igual ao da pesquisa por gene, no entanto vamos mostrar um tipo de resultado diferente uma vez que como na figura 5.22, temos resultados onde os valores de início e fim das sequências estão invertidos e inicialmente, na figura 5.21, escolhemos ver os rRNA 5s e rRNA 16s. Veremos na figura 5.23, uma amostra do resultado obtido dos gráficos circulares provenientes do ficheiro XML resultante do alinhamento do BLAST. Relembramos que cada tipo de RNA tem uma cor diferente e são representados por linhas mostrando as suas suas posições nos respectivos organismos.

Gráficos Lineares

O objectivo destes gráficos é mostrar todos os genes existentes em cada organismo em forma de setas orientadas tendo como referência o início e fim de cada sequência encontrada para cada organismo. Para a realização deste modo de vista necessitamos com base nos dados do ficheiro XML em 5.13 a informação relativa ao nome de cada organismo (*Hit_def*, o início (*Hsp_hit-from*) e o fim (*Hsp_hit-to*) de cada sequência desse organismo. Com base na informação inserida na base de dados MySQL necessitamos de saber quais os genes (*srt* e *stp* de cada um), o *locus* e o *complement* encontrados para

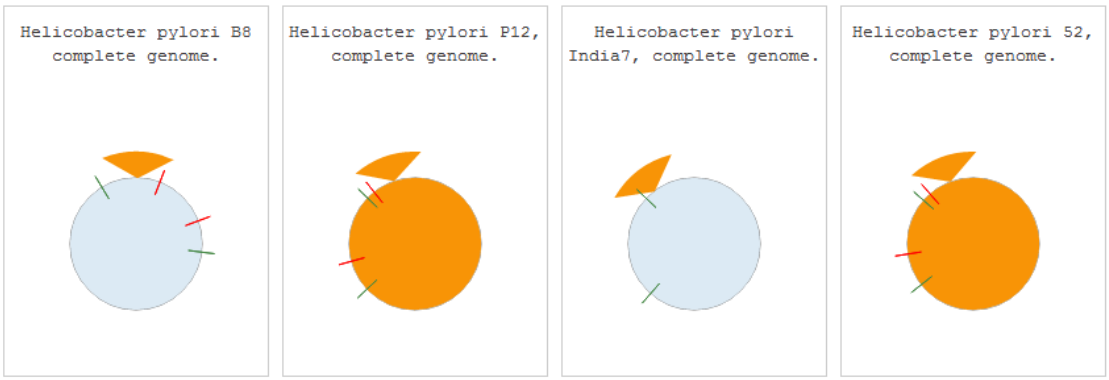


Figura 5.23: Gráficos circulares da pesquisa por região com tipo de rRNA 5s e rRNA 16s. *(http://xldb.fc.ul.pt/biotools/gin/)*

cada organismo existente no ficheiro XML compreendidos no intervalo de início e fim da sequência encontrada.

É desenhado no lado do cliente através do elemento *canvas* com HTML5 e JavaScript uma linha de cor preta, essa mesma linha vai corresponder ao tamanho da sequência do organismo em questão, sendo o tamanho dessa sequência igual ao intervalo compreendido entre o início e o fim da mesma, ou seja, sabemos que o intervalo começa em *Hsp_from* e acaba em *Hsp_to* e esses valores são apresentados abaixo da linha. São também desenhadas setas correspondentes a cada gene encontrado no intervalo da sequência obtida pelo alinhamento do BLAST. As setas são desenhadas da esquerda para a direita se o resultado do seu complemento (complement) for "N" e da direita para a esquerda se for "y". A cada seta é atribuída uma cor consoante o tamanho (*stp* - *srt*) da mesma como podemos ver na figura 5.24 Estes passos são repetidos enquanto houver blocos de resultados por parte do ficheiro XML que obtivemos através do BLAST.

Alinhamentos múltiplos

Os alinhamentos múltiplos funcionam exactamente da mesma forma que os acima referidos. É criada uma tabela referente aos dados provenientes do ficheiro XML que o BLAST devolveu em 5.13, e as sequências pré preenchidas ou alteradas pelo utilizador vão ser retiradas da base de dados FASTA com o nome escolhido inicialmente na opção *Data base* e escritas num ficheiro temporário de maneira a que o MUSCLE seja executado sobre esse ficheiro. Escolhe-se o formato de saída do MUSCLE e fazemos *Muscle it*. Podemos mostrar um outro formato de saída do alinhamento múltiplo (ClustalW), o alinhamento da figura 5.25, é feito com base nos dados acima escolhidos e sobre os 6 primeiros organismos.

Figura 5.24: Gráficos lineares da pesquisa por região.
(http://xldb.fc.ul.pt/biotools/gin/)

5.5.3 Pesquisa por sequência

Na pesquisa por sequência o raciocínio é praticamente igual às outras opções, a grande diferença é que não existe a opção de escolha de organismo (*Kernel genome*) pois a pesquisa é feita com base numa sequência introduzida manualmente pelo utilizador ou através de um ficheiro externo como podemos verificar na figura 5.26

Os passos necessários para a realização da pesquisa por sequência passam por escolher qual a base de dados que se pretende na opção *Data Base*, neste momento será mostrado na opção *Genomes* e todos os genomas correspondentes a essa base de dados escolhida. Após escolher a base de dados selecciona-se uma das seguintes opções.

```
Genome_nº_6      ---------CAAGTCAAAGATGATAAAAAATTTCTCTCATCAAAAATTTCATCGCA--TA
Genome_nº_4      ---------CAAGTCAAAGATGATAAAAAATTCCCCTCCCCAAAAATTTCATCGCA--CA
Genome_nº_2      ---------CAAGTCAAAGATGATAAAAAATTCCCCTCCCCAAAAATCTCATCGCA--CA
Genome_nº_5      -------------------------AAAATTCCCCTCCCCAAAAATTTCATCGCA--TA
Genome_nº_1      GAAAGTGTGCGAAATAAAGACAGAAAAAAG-----CCTTACAAAGGCTGAGTCGTATTCA
Genome_nº_3      GAAAGTGTGCGAGATAAAGACAGAAAAAAG-----CCTTACAAAGGGTGAGTTGTATTCA
                                          ***      *    ****       * * *   *

Genome_nº_6      AAAGTTTGAGTTGGGCGACTTCATTATCG--TCAATGCTGATAAAAATCACGCCGTCTTG
Genome_nº_4      AGATTTTTAAATTCGCGCATTCATTATCG--TCAATGCTGATGAAAATCACGCCGTCTTG
Genome_nº_2      AGATTTTTAAATTCGCGCACTCGTTATCG--TCAATGCTGATGAAAATCACGCCGTCTTG
Genome_nº_5      AAAGTTTGAGTTGGGCGGCTTCGTTATCG--TCAATAGAAATGAAAATCACGCCGTCTTG
Genome_nº_1      AAAGC----AAAAGGCCATTTGATCCCCACTTCAATACTAG------------CGTCATG
Genome_nº_3      AAAGC----AAAGGGCCATTTGACTCCCACCTCAATACTAG------------CGTCATG
                  * *           **    *      *   *****                 **** **

Genome_nº_6      TTTGAGCAAATCTTTAGCGAGCAACAATCTAGGATACATGAAACTAAGCCACCCGCTATG
Genome_nº_4      TTTGAGCAAATCTTTAGCGAGCAACAATCTAGGATACATGAAACTAAGCCACCCGCTATG
Genome_nº_2      TTTGAGCAAATCTCTAGCGAGTAGCAATCTAGGATACATGAAACTAAGCCACCCGCTATG
Genome_nº_5      TTTGAGCAAATCTCTAGCGAGCAGCAATCTAGGATACATGAAACTAAGCCACCCGCTATG
Genome_nº_1      GTGGAA--------TAGC-----ACGGTTTGGAAT---TTAAAC--------CCCACTA--
Genome_nº_3      GTGGAA--------TAGC-----ACAGTTTGGAAT---TTAAAC--------CCCACTA--
                   * **         ****      *  * * * **   * ****        *** ***
```

Figura 5.25: Resultado do MUSCLE da pesquisa por região.
(http://xldb.fc.ul.pt/biotools/gin/)

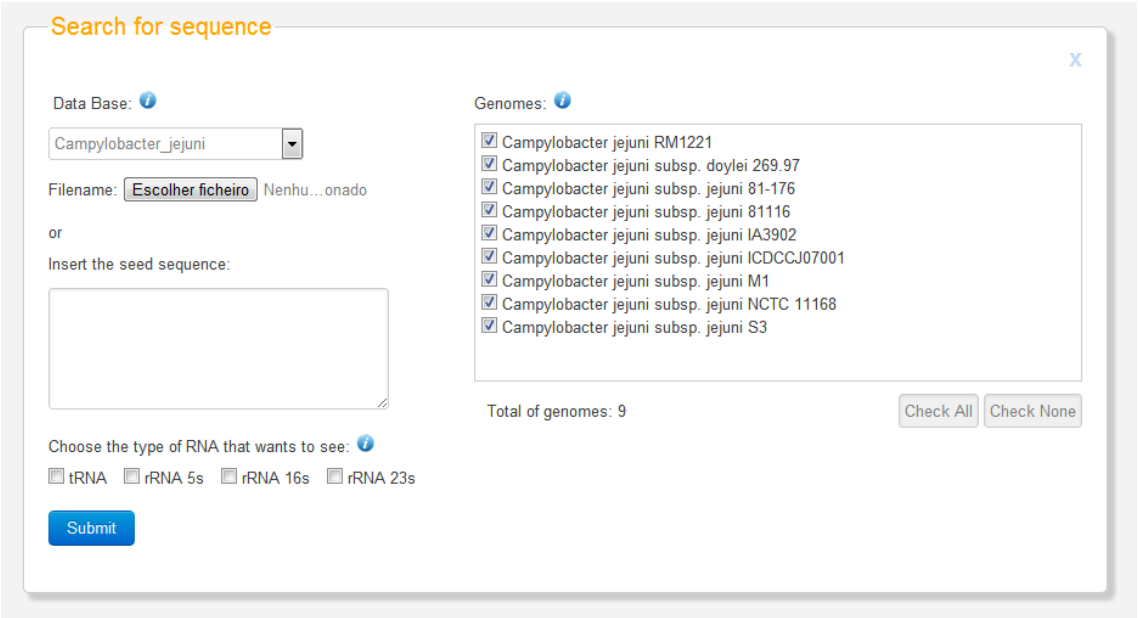


Figura 5.26: Pesquisa por sequência.
(http://xldb.fc.ul.pt/biotools/gin/)

- O utilizador insere a sua sequência manualmente na opção *insert the seed sequence*.
- O utilizador faz *upload* de um ficheiro externo com a sua sequência, de modo a ser possível "correr" essa sequência contra os genomas escolhidos pelo utilizador na opção *Genomes*.

Escolhe-se quais os genomas pretendidos na opção *Genomes* para efectuar o alinha-

mento de sequências através do BLAST. Escolher ou não algum/alguns tipos de RNA que se pretende ver nos gráficos circulares e por fim submeter os dados através do botão *Submit*. Após os dados serem submetidos, primeiro é verificado qual das opções (ficheiro externo com a sequência ou inserir a sequência manualmente) foi escolhida pelo utilizador, depois então é que os dados são enviados para o lado do servidor e deste modo proceder à recolha de informação necessária para executar o BLAST e termos o seu resultado em XML para o pudermos trabalhar.

Para obtermos o resultado do BLAST em formato XML como em 5.13 primeiro verifica-se qual a opção de inserção da sequência que o utilizador optou.

- Se o utilizador introduziu manualmente a sua sequência esta é guardada num ficheiro temporário em formato FASTA para depois ser utilizado com o BLAST.

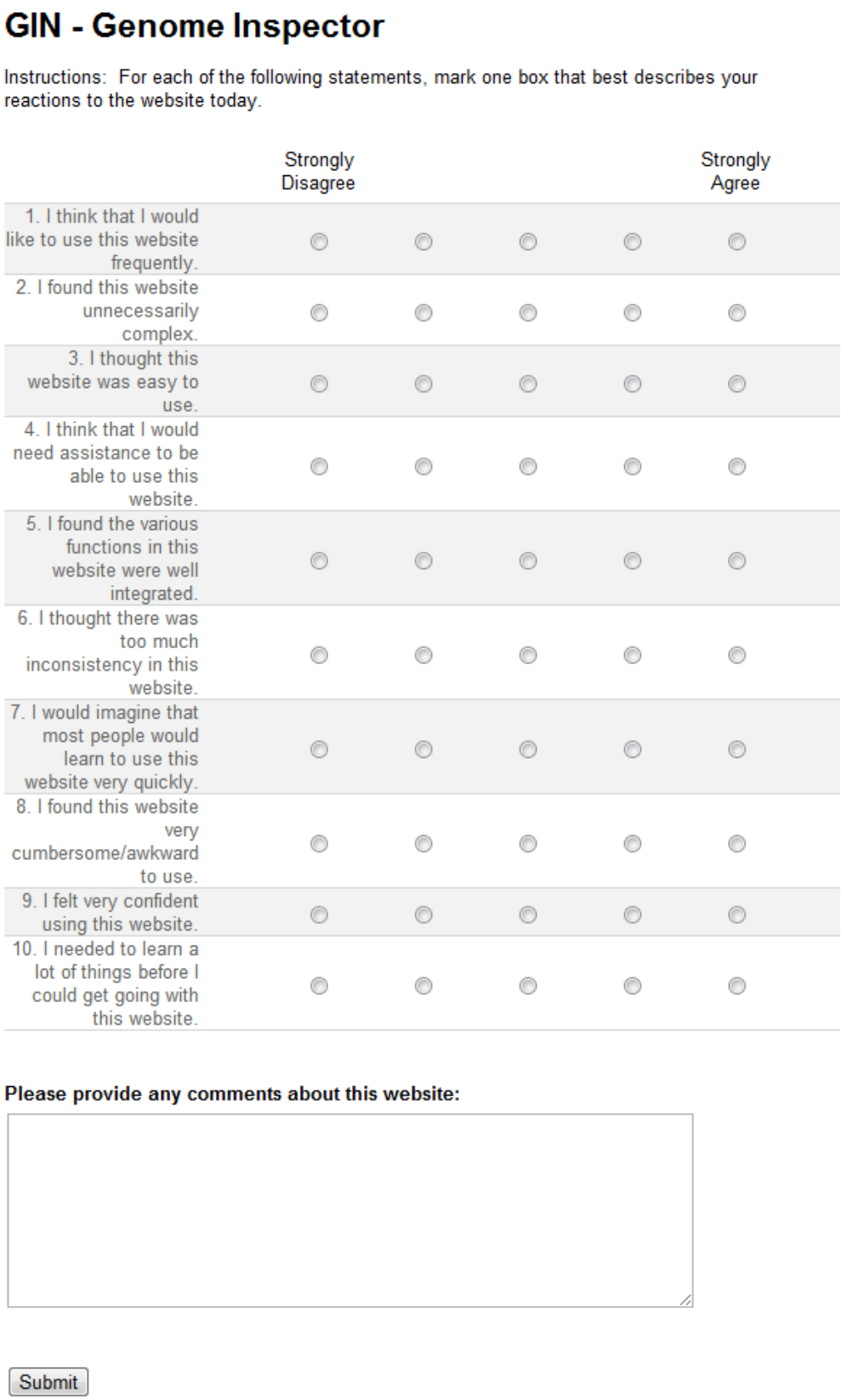
- Se o utilizador escolheu carregar a sua sequência através de um ficheiro externo, primeiramente recolhemos o conteúdo do mesmo, caso o ficheiro seja menor que 2MB, e enviamos o conteúdo para o lado do servidor.

  - Se o conteúdo começar com o caractere > o mesmo é escrito para um ficheiro temporário em formato FASTA.
  - Se o conteúdo não começar com > então escrevemos para um ficheiro temporário uma linha a começar com > e com uma descrição e nas linhas seguintes o conteúdo.

É então executado o BLAST sobre o ficheiro criado anteriormente com a sequência e sobre a base de dados FASTA como nome da base de dados escolhida inicialmente. O resultado é dado num ficheiro com o formato XML. O resultado das opções de vista é obtido a partir desse ficheiro XML através do alinhamento do BLAST. Todos os modos de vista desta opção já foram explicados acima nas outras opções de vista, uma vez que o modo e os métodos utilizados foram os mesmos.

5.6 Testes de usabilidade

Os testes de usabilidade para todos os utilizadores ainda estão a decorrer, de modo que ainda não temos material suficiente para mostrar gráficos ou estatísticas referentes aos testes de usabilidade. O modelo que usámos para realizar os nossos testes foram 10 perguntas que permitem saber se a aplicação está bem estruturada a nível de *design*, se está perceptível e se realmente é útil e funcional. As perguntas têm uma escala de concordância que permite uma avaliação distinta de utilizador para utilizador. A figura 5.27 mostra então o modelo que usamos para a realização dos nossos testes de usabilidade.



GIN - Genome Inspector

Instructions: For each of the following statements, mark one box that best describes your reactions to the website today.

	Strongly Disagree				Strongly Agree
1. I think that I would like to use this website frequently.					
2. I found this website unnecessarily complex.					
3. I thought this website was easy to use.					
4. I think that I would need assistance to be able to use this website.					
5. I found the various functions in this website were well integrated.					
6. I thought there was too much inconsistency in this website.					
7. I would imagine that most people would learn to use this website very quickly.					
8. I found this website very cumbersome/awkward to use.					
9. I felt very confident using this website.					
10. I needed to learn a lot of things before I could get going with this website.					

Please provide any comments about this website:

Submit

Figura 5.27: Testes de usabilidade.
(https://docs.google.com/spreadsheet/viewform?formkey=dER4ZE5vVDVzb0NDNFVDNFZUMk5NZkE6MQ#gid=0)

Capítulo 6

Conclusão

O GIN permite que um utilizador possa inserir os seus genomas completos através de um pedido ao administrador e que estes possam estar alojados por um período de tempo finito. Assim cada utilizador pode fazer as suas pesquisas de sequências genéticas através da sua própria base de dados. O utilizador será avisado por *email* de quanto tempo a sua base de dados estará disponível e passado esse período a mesma será apagada. O GIN foi criado de raiz sendo por isso possível explorar todos os conceitos adquiridos ao longo do nosso curso e coloca-los em prática na realização da nossa aplicação e de todo o sistema envolvente. Tivemos o cuidado de criar um sistema de informação recorrendo a processos assíncronos de modo a que o utilizador não se disperse durante a navegação e de modo a que todos os dados sejam alterados e mostrados sem ser necessário actualizar toda a página *web*.

É dada a possibilidade de o utilizador após escolher o método de pesquisa de uma determinada sequência genética, poder escolher quais os modos de vista que pretende visualizar, entre os quais, o alinhamento simples, gráficos circulares, gráficos lineares e alinhamentos múltiplos. Para as vistas dos gráficos foi utilizada a mais recente versão do HTML, versão 5, com o uso da *tag canvas*, tentando explorar algumas das mais recentes tecnologias *web*. A desvantagem é que nem todos os *browsers* estão preparados para receber tal tecnologia, no entanto caso isso aconteça será mostrado um aviso.

Como observações finais podemos concluir que a realização deste projecto serviu para a resolução de vários problemas existentes na área de informação genética microbiana, como por exemplo o facto de o utilizador ter acesso a informações específicas sobre uma determinada sequência ou gene e ser possível comparar num determinado modo de vista, vários resultados de alinhamentos ou de gráficos genómicos, circulares ou lineares.

6.0.1 Desafios encontrados

Uma das dificuldades encontradas foi perceber e entender todo o problema à volta da visualização de mapas genómicos microbianos, e de como poderíamos apresentar os resultados pretendidos, de forma a que fosse perceptível para o utilizador. O facto de o

HTML5 ser recente, tornou-se um desafio aprender um pouco mais sobre esta nova versão e de como poderíamos usar as suas funcionalidades em prol do nosso projecto.

A utilização de novas linguagens como o Python e Biopython para a resolução dos problemas de alinhamento de sequências foi bastante proveitoso devido à eficiência das ferramentas disponíveis pelo módulo BioPython, no entanto foi necessário perceber e compreender toda a estrutura da linguagem Python.

6.0.2 Trabalho futuro

Como trabalho futuro prevê-se que seja optimizada a base de dados *gindb*, que seja incluída na mesma todo o genoma completo de cada organismo e desta feita já não seria necessário pesquisar dentro da base de dados FASTA uma determinada sequência de um organismo. Já essa pesquisa seria feita directamente na base de dados MySQL.

Outras das alterações interessantes seria ter um *script* que verificasse em que dia foi criada uma base de dados para um determinado utilizador e qual a duração da mesma. De x em x dias seria feita uma verificação do período determinado pelo administrador em que a base de dados estaria disponível e alojada, caso esse período fosse ultrapassado um *email* seria enviado ao utilizador em questão a avisar que a sua base de dados seria eliminada. A implementação de segurança com HTTPS também seria interessante introduzir no GIN de modo a termos uma navegação e troca de informação mais segura. Ainda a nível de segurança o tratamento de *SQL injection* também poderia ser implementado de forma a que toda a comunicação entre o servidor e a base de dados relacional MySQL fosse feita de forma segura.

A nível de compatibilidade de *browser* com o HTML5, caso um determinado *browser* não consiga interpretar a *tag canvas* ser possível ter uma alternativa à mostragem gráfica dos mapas genómicos.

Bibliografia

[1] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215:403–410, 1990.

[2] Dennis a Benson, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and Eric W Sayers. GenBank. *Nucleic acids research*, 37:D26–31, January 2009.

[3] Dennis a Benson, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and David L Wheeler. GenBank. *Nucleic acids research*, 36:D25–30, January 2008.

[4] C Camacho, Thomas Madden, George Coulouris, and N Ma. BLAST Command Line Applications User Manual. 1997, 2008.

[5] Hugh Darwen. *An Introduction to Relational Database Theory*, volume 39. BookBoon.com, 2010.

[6] Chrisina Draganova. Asynchronous JavaScript Technology and XML (AJAX). *Technology*, pages 3–10, 2007.

[7] By Paul Dubois and Pub Date. MySQL Cookbook. *Database*, (October):1022, 2002.

[8] D Eastlake 3rd and P Jones. US Secure Hash Algorithm 1 (SHA1), 2001.

[9] Robert C Edgar. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC bioinformatics*, 5:113, 2004.

[10] Robert C Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32:1792–1797, 2004.

[11] R T Fielding and G Kaiser. The Apache HTTP Server Project, 1997.

[12] R D Fleischmann, M D Adams, O White, R A Clayton, E F Kirkness, A R Kerlavage, C J Bult, J F Tomb, B A Dougherty, J M Merrick, and Et Al. Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science*, 269:496–512, 1995.

[13] Paulien Hogeweg. The Roots of Bioinformatics in Theoretical Biology. *PLoS Computational Biology*, 7:5, 2011.

[14] Peter Lubbers, Brian Albers, and Frank Salim. Overview of HTML5. In *Pro HTML5 Programming*, pages 1–23. Apress, 2010.

[15] David W. Mount. *Bioinformatics: sequence and genome analysis*. John Inglis, 2nd ed. edition, 2004.

[16] Richard J Roberts, Tamas Vincze, Janos Posfai, and Dana Macelis. REBASE–a database for DNA restriction and modification: enzymes, genes and genomes. *Nucleic acids research*, 38(Database issue):D234–6, January 2010.

[17] Christopher Schmitt. *CSS Cookbook*, volume 22. O'Reilly, 2006.

[18] David B Searls. The roots of bioinformatics. *PLoS computational biology*, 6:7, June 2010.

[19] BLAST. URL: `http://blast.ncbi.nlm.nih.gov/`, July 2012.

[20] FASTA. URL: `http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml`, July 2012.

[21] GenBank. URL: `http://www.ncbi.nlm.nih.gov/genbank/`, July 2012.

[22] GIN. URL: `http://xldb.fc.ul.pt/biotools/gin/`, August 2012.

[23] MUSCLE. URL: `http://www.drive5.com/muscle/`, July 2012.

[24] REBASE. URL: `http://rebase.neb.com/rebase/rebase.html`, July 2012.

[25] By David Sklar and Adam Trachtenberg. *PHP Cookbook*. Number March 2008 in Cookbooks Series. O'Reilly, 2002.

[26] David Wheeler and Medha Bhagwat. BLAST QuickStart: example-driven web-based BLAST tutorial. *Methods in molecular biology Clifton NJ*, 395:149–176, 2007.